

EÖTVÖS LORÁND TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI KAR

NEURÁLIS HÁLÓK

MSc Szakdolgozat

Sáfár Rebeka

Alkalmazott matematikus MSc,
Sztochasztika szakirány

Témavezető:

Prőhle Tamás,

egyetemi tanársegéd

Valószínűségelméleti és Statisztika Tanszék,
Eötvös Loránd Tudományegyetem



Budapest
2014

Tartalomjegyzék

1. Bevezetés	1
2. A neurális hálók elméletének alapfogalmai	2
2.1. A perceptron	3
2.2. A többétegű perceptron	4
3. Kolmogorov szuperpozíciós tétele	8
4. A szuperpozíciós tételben szereplő ψ függvény előállítása D. A. Sprecher nyomán	11
5. A szuperpozíciós tétel és a neurális hálók kapcsolata	17
5.1. A differenciálhatóság problémája	20
6. Függvényapproximáció szigmoid függvények segítségével	23
6.1. A $\mathcal{C}(\mathcal{E}^n)$ -beli sűrűség	23
6.2. Neurális hálók alkalmazása	25
6.3. Más aktivációs függvények	27
7. Az R program neurális hálókkal foglalkozó csomagjainak bemutatása	31
7.1. Az nnet	31
7.2. A neuralnet	33
7.2.1. Ábrázolás	35
7.3. Az RSNNS	37
7.3.1. Ábrázolás	40
8. A neuralnet használata életminőségi adatokra	41
8.1. Az R-ben futtatott függvények és az általuk kapott eredmények	45
8.2. Az adatbázis	49
Irodalomjegyzék	50

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani a témavezetőmnek Próhle Tamásnak, aki rengeteg időt és energiát fordított rám és a munkámra. Folyamatos türelmével, segítségével és tanácsaival nagyban segítette a dolgozat megszületését. Köszönet illeti továbbá Boruzs Gergelyt a dolgozat elkészítése közben felmerülő informatikai problémák megoldásában nyújtott segítségéért és megértéséért valamint hálával tartozom a szüleimnek a sok támogatásért és bátorításért, amit a tanulmányaim során kaptam tőlük.

1. fejezet

Bevezetés

David Hilbert 1900 augusztusában a párizsi Nemzetközi Matematikai Kongresszuson felvázolta azt a 23 matematikai problémát, melyet a huszadik század legjelentősebbjeinek tartott. A tizenharmadik probléma a függvények kompozíciójáról szól, speciálisan arról, hogy felírhatók-e magasabb fokú egyenletek alacsonyabb fokú vagy egyváltozós egyenletek kompozíciójaként.

Most lássuk a tizenharmadik problémát pontosan:

Megoldhatók-e az

$$x^7 + ax^3 + bx^2 + x + 1 = 0$$

alakú hetedfokú egyenletek véges sok két változós folytonos függvény alkalmazásával?

Ennek általánosítására adott egy eredményt Kolmogorov 1957-ben, amikor kimondta a következő tételt:

1.0.1. Kolmogorov szuperpozíciós tétele. *Minden $n \geq 2$ változós f függvény reprezentálható valós értékű $\phi^{p,q}(x)$, a zárt $\mathcal{E}^1 = [0; 1]$ intervallumon folytonos függvényekkel, a következő alakban:*

$$f(x_1, x_2, \dots, x_n) = \sum_{q=1}^{q=2n+1} \chi_q \left[\sum_{p=1}^n \psi^{p,q}(x_p) \right] \quad (1.1)$$

ahol $\chi_q(y)$ folytonos, valós értékű függvény.

A tételt a második fejezetben tárgyaljuk. Az ezt követő fejezetekben bemutatjuk a tételhez kapcsolódó eredményeket, majd megmutatjuk, hogy ez a tétel hogyan kapcsolódik a neurális hálókhoz és azt is, hogy hogyan tudunk ezekkel a hálókkal függvényeket közelíteni. Később beszélünk arról is, hogy a neurális hálók területén elért eredmények szempontjából Kolmogorov tétele mennyire volt releváns. Ahhoz, hogy mindezt könnyebben végrehajthassuk az első fejezetben bevezetjük a neurális hálók elméletének alapvető fogalmait. Az utolsó fejezetekben pedig konkrét példákon mutatjuk be az R program neurális hálókkal foglalkozó csomagjait.

2. fejezet

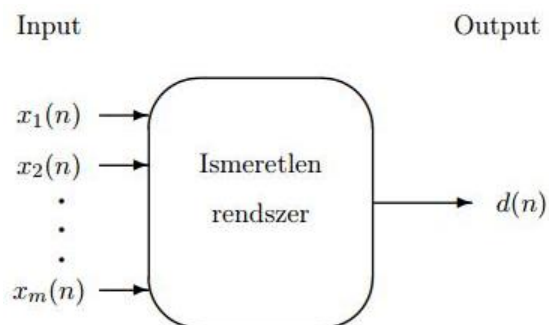
A neurális hálók elméletének alapfogalmai

Ebben a fejezetben a neurális hálók elméletének azon fogalmait vázoljuk fel, melyekre a dolgozat további fejezeteiben szükségünk lesz.

A neurális hálókat egyre szélesebb körben alkalmazzák olyan műszaki és tudományos feladatok megoldására, melyek visszavezethetők két egyszerű problémára: az osztályozásra és a függvényközelítésre. Ilyenek például a kép feldolgozás, adatbányászat, karakterfelismerés és különböző bioinformatikai problémák.

A biológiai rendszerek mintájára mesterséges rendszereket próbáltak létrehozni. A tervezés során olyan alapelveket tartottak szem előtt, mint az sok apró egységből (neuronból) álló felépítés, az egységek közötti kapcsolatok és a tanulásra való képesség. A biológiai rendszerek mélyebb tulajdonságaival viszont a mesterséges neurális hálózatok nem rendelkeznek. Ezek alapján Fazekas Istvánt idézve csak egy közelítő meghatározást adhatunk, miszerint *a mesterséges neurális hálózat az idegrendszer felépítése és működése analógiájára kialakított számítási mechanizmus*.

Vannak olyan műszaki problémák, ahol nem támaszkodhatunk törvényekre, vagy használható modellekre, viszont rendelkezünk mérési adatokkal vagy megfigyelésekkel és ezekkel kell használható eredményeket elérnünk. Ilyen esetekben a jelenséget magát egy fekete dobozként kezeljük.



A jó neurális háló kimeneti adatai hasonlítanak a jelenség kimeneti adataira, a mechanizmust magát pedig nem tárja fel (ilyenek például a nem-paraméteres statisztikai módszerek is). Ezt láthatjuk az előző ábrán. Így a neurális hálózatok működéséhez csak adatokra van szükség.

A neurális hálók legelterjedtebb típusa az egy neuronból álló perceptron volt. Később azonban rájöttek, hogy ezzel több neuront egy rétegbe rendezve sem oldható meg lineárisnál bonyolultabb feladat. Majd jött a többrétegű perceptron és a betanítására alkalmas eljárás a hiba visszaáramoltatásának algoritmus (error backpropagation). Ez igazi áttörést jelentett, azóta a neurális hálók elmélete és az alkalmazási területek hatalmas fejlődésen mentek át.

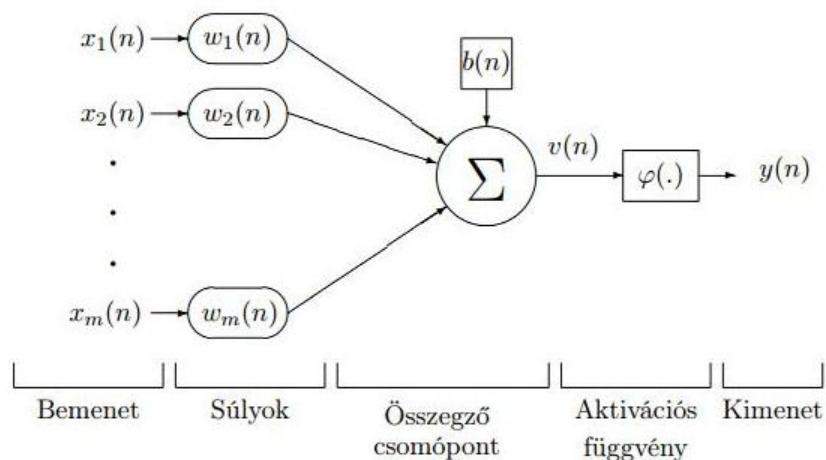
A többrétegű perceptron, tehát a legelterjedtebb a neurális hálók között, rétegekbe rendezett perceptronokból áll. Ez egy előrecsatolt hálózat, vagyis az input jel rétegről rétegre halad előre (ha úgy tekintünk a hálózatra mint egy gráfra, akkor abban nem lesz kör és hurok sem).

A neurális hálózatok tanítása rendkívül fontos. Megkülönböztetünk irányított (supervised) és irányítás nélküli (unsupervised) tanulást, mi az előbbiről, tehát az irányított tanulásról beszélünk bővebben.

Az irányított tanulásnál rendelkezünk tanító pontokkal (training points), tehát adott $x(1), \dots, x(N)$ input adatokhoz ismerjük a $d(1), \dots, d(N)$ output értékeket is. Ezek alapján illeszthetjük a modellünket, tehát olyan paraméter értékeket választunk, melyek mellett egy adott inputra hasonló eredményeket kapunk, mint amelyet a tanító pontoknál látunk. A jól betanított hálót ezek után használhatjuk olyan adatokra, melyeknek nem ismerjük még az outputját.

2.1. A perceptron

A következőkben a perceptron felépítését és működését mutatjuk be részletesebben.



- Az ábrán látható bemenet, vagyis input elemei ($\mathbf{x}(n) = x_1(n), \dots, x_m(n)$) ismertek. Ez az m -dimenziós vektor az n -dik időpontban érkezik (minden $n = 1, 2, \dots, N$ időpontban érkezik egy bemenő jel).
- A w_1, \dots, w_m súlyok nem ismertek. Az ábrán látható $w_1(n), \dots, w_m(n)$ értékek az n -edik időpillanatban használt súlyok. Ezek az igazi w_1, \dots, w_m súlyok n -edik közelítései (a közelítéseket lépésenként tudjuk finomítani). Mind az igazi, mind a közelítő súlyokat vektorokként kezeljük majd, úgy ahogy a bemenetnél is láthattuk.
- Az ábrán látható $b(n)$ az igazi torzítás (amit nem ismerünk) n -edik közelítését jelöli (b általában skalár). Célunk a súlyvektor és a torzítás meghatározása.
- A következő állomás, amit láthatunk az összegző csomópont. Ezt a bemenet adatokból kaphatjuk meg az alábbi összefüggés alapján:

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

- Az aktivációs függvény, melyet $\varphi(\cdot)$ -vel jelölünk, az alkalmazási területtől és a megoldandó problémától függ és a fenti $v(n)$ összeget alakítja át. A megfelelő aktivációs függvényt szintén nekünk kell megadnunk.
- Végül megkapjuk a kimenetet (más szóval outputot). Ez a bemeneti $\mathbf{x}(n)$ értékhez a neuron által rendelt érték, tehát $y(n) = \varphi(v(n))$.

Az egyszerűsítés végett kiegészítjük a súlyok és a bemenő jelek vektorát egy nulladik koordinátával, speciálisan a $x_0(n) = 1$ és a $w_0(n) = b(n)$ értékekkel. Tehát a innentől a torzítást is súlyként kezeljük. Ezután a tanító pontokra alkalmazva a neurális hálót összehasonlítjuk a kapott $y(n)$ értékeket a már ismert $d(n)$ értékekkel, így kapjuk meg a hálózat négyzetes hibáját:

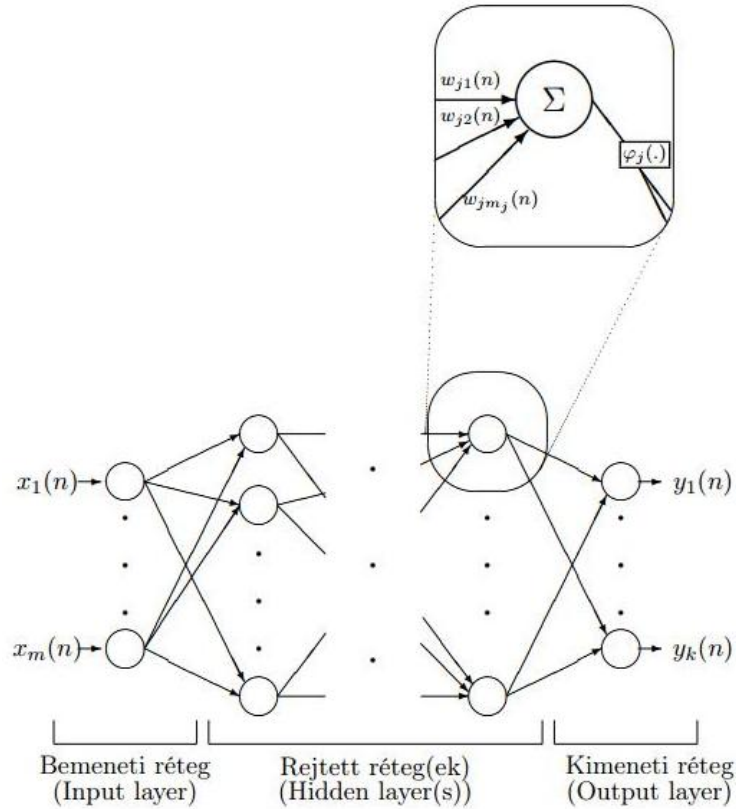
$$E(n) = (d(n) - y(n))^2.$$

A célunk ennek az értéknek a minimalizálása, ezt a súlyok változtatásával tudjuk elérni.

2.2. A többétegű perceptron

Eddig a perceptront mutattuk be, a következőkben pedig a többétegű perceptron felépítéséről és tanításáról lesz szó. Itt nem egyszerűen neuronokat kapcsolunk össze egy hálózatban, hanem rétegekbe is rendezzük őket. A rétegeket három csoportba sorolhatjuk: bemeneti réteg (input layer), rejtett rétegek (hidden layers) és output réteg. A rejtett rétegek száma tehát tetszőleges lehet, míg az input és output rétegekből csak egy van.

A következő ábrán egy előrecsatolt hálózat (feedforward network) látható, amelyben a jel balról áramlik jobbra, vagyis egy adott rétegen belüli neuron bemenete a tőle közvetlenül balra lévő neuron kimenete. Ebben a modellben minden neuron kapcsolatban van a vele



közvetlenül szomszédos rétegek minden neuronjával. Fontos megjegyeznünk azt is, hogy minden neuronnak saját aktivációs függvénye és saját súlyai vannak.

A következőkben rátérünk a többrétegű perceptron tanítására. A számos módszer közül mi most egyet mutatunk be részletesebben, mégpedig a hiba visszaáramoltatásának módszerét (error backpropagation algorithm). A tanulást a következő főbb lépésekre bont-hatjuk:

- megadjuk a kezdeti súlyokat,
- a tanító pontot végigáramoltatjuk az egész hálózaton,
- az így kapott jelek valamint a tanító pontokhoz tartozó kimeneti jelek alapján kiszá-moljuk a hibát,
- a kapott hibát visszaáramoltatjuk az egész hálózaton, a súlyokat pedig úgy változ-tatjuk meg, hogy a hiba csökkenjen.

Ahhoz, hogy ezt részletesebben láthassuk először be kell vezetnünk a következő jelöléseket:

i, j, k : i -edik, j -edik és k -adik neuron;

n : a tanítás n -edik lépése;

$y_i(n)$: az i -edik neuron kimenete és az őt közvetlenül követő neuron bemenete;

$y_0(n) \equiv 1$;

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

$w_{ji}(n)$: az i -edik neuronból a j -edik neuronba vezető él súlya;

$v_j(n)$: a következő összeg

$$v_j(n) = \sum_i w_{ji}(n) y_i(n),$$

ahol az i szerinti összegzés azt jelenti, hogy a j -edik réteg előtti rétegeken haladunk előre;

$\varphi_j(\cdot)$: a j -edik neuron aktivációs függvénye;

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

$d_j(n)$: az output valódi értéke (csak a kimeneti rétegben ismert).

Ezekkel a jelölésekkel felírhatjuk az n -edik lépés hibáját a következőképpen:

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

ahol \mathcal{C} az output réteg neuronjait jelöli. Ebből az átlagos négyzetes hibát is kiszámolhatjuk úgy, hogy az $\mathcal{E}(n)$ értékeket összegezzük 1-től N -ig, majd a kapott összeget N -nel vagyis a tanító pontok számával osztjuk el. Itt megjegyezzük, hogy bár ez a legelterjedtebb hibafüggvény, de ehelyett másfajta függvényeket is használhatunk, mint például a kereszt-entrópia függvény (cross-entropy)

A súlyokat a gradiens módszer vagyis a *delta rule* segítségével változtathatjuk meg:

$$w_{ji}(n+1) - w_{ji}(n) = \Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}, \quad (2.1)$$

ahol η a tanulási paraméter ($\eta > 0$). A tanulási paraméter az algoritmus sebességét és stabilitását szabályozza. Kicsi η esetén csak kicsit változnak a súlyok, ezáltal lassú a konvergencia. Nagy η esetén gyorsabb lesz a konvergencia, de az algoritmus elvesztheti a stabilitását és oszcillálóvá válhat. A tanulási paramétert menet közben is megváltoztatjuk.

Az error backpropagation algoritmus rekurzívan számítja ki a deriváltakat: a j -edik neuron lokális gradiensét megkapjuk a j utáni réteg lokális gradienseiből és mivel az utolsó réteg lokális gradienseit közvetlenül kiszámíthatjuk, így végül az összes lokális gradienst megkaphatjuk. A lokális gradienst a következő összefüggés határozza meg:

$$\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = e_j(n) \varphi'_j(v_j(n)). \quad (2.2)$$

Ezt a (2.1) összefüggésből kaphatjuk meg az összetett függvény deriválási szabályának alkalmazásával. Ha a j -edik neuron a kimeneti rétegben van, akkor az előző összefüggés egyszerűen ad eredményt, mivel $e_j(n)$ -t megkaphatjuk $d_j(n)$ és $y_j(n)$ különbségeként. Abban az esetben, ha a j -edik neuron egy rejtett rétegben van, akkor kicsit tovább kell számolnunk.

$$\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = -\sum_k \frac{\partial \mathcal{E}(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial v_j(n)} = \sum_k \delta_k(n) w_{kj}(n) \varphi'_k(v_k(n)),$$

az összetett függvény deriválási szabálya miatt, ahol k a j -edik neuron rétege utáni rétegben van. Így a többrétegű perceptron szerkezete miatt

$$v_k(n) = \sum_l w_{kl}(n) \varphi_l(v_l(n)),$$

ahol az l -edik neuron a j -edik neuron rétegében szerepel. Ha ezt $v_j(n)$ szerint deriváljuk, $w_{kj}(n)\varphi'_j(v_j(n))$ -t kapunk eredményül, ezzel pedig azt kapjuk, hogy

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n), \quad (2.3)$$

ahol k a j utáni rétegben szerepel. Az előzőeket foglalja össze a következő formula:

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

A súlyokat tehát a következő összefüggés szerint változtatjuk meg:

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

ahol $y_i(n)$ jelöli a j -edik neuron rétege előtt lévő rétegben szereplő i -edik neuron outputját, $\delta_j(n)$ -et pedig a fenti összefüggések határozzák meg.

Az algoritmus megállítására a következő szabályszerűségek adhatók:

- ha a gradiens vektor kicsi,
- ha egy epoch alatt a négyzetes hiba csak kis mértékben csökken (egy epochnak nevezzük azt, ha az összes tanító pontot egyszer átáramoltatjuk az egész hálón),
- ha az epochok száma vagy a futási idő túl nagy, akkor megállunk.

Az error backpropagation algoritmus érzékeny a kezdeti értékekre, sőt az is előfordulhat, hogy csak egy lokális minimumot találunk meg vele, tehát nagyon fontos a kezdeti súlyok megválasztása.

3. fejezet

Kolmogorov szuperpozíciós tétele

Most rátérünk A. N. Kolmogorov 1957-es eredményére, mely a függvények kompozíciójáról szól.

3.0.1. Tétel. Minden $n \geq 2$ változós f függvény reprezentálható valós értékű $\phi^{p,q}(x)$, a zárt $\mathcal{E}^1 = [0; 1]$ intervallumon folytonos függvényekkel, a következő alakban:

$$f(x_1, x_2, \dots, x_n) = \sum_{q=1}^{q=2n+1} \chi_q \left[\sum_{p=1}^n \psi^{p,q}(x_p) \right] \quad (3.1)$$

ahol $\chi_q(y)$ folytonos, valós értékű függvény.

Ahhoz, hogy ezt belássuk először megfelelő $\psi^{p,q}$ függvények konstruálunk, olyan p, q, k és i értékekre, melyekre a következők teljesülnek: $1 \leq p \leq n$, $1 \leq q \leq 2n+1$, $k = 1, 2, \dots$ és $1 \leq i \leq m_k = (9n)^k + 1$. Tekintsük az

$$A_{k,i}^q = \left[\frac{1}{(9n)^k} \left(i - 1 - \frac{9}{3n} \right), \frac{1}{(9n)^k} \left(i - \frac{1}{3n} - \frac{q}{3n} \right) \right]$$

zárt intervallumokat, melyek hossza $\frac{1}{(9n)^k} \left(1 - \frac{1}{3n} \right)$ rögzített k és q mellett. Ezek az $A_{k,i}^q$ intervallumok lefedik az egész \mathcal{E}^1 intervallumot. Tehát fix k -ra és q -ra kapott $S_{k,i_1 \dots i_n}^q = \prod_n A_{k,i_p}^q$ n -dimenziós kockák az egész \mathcal{E}^n -et lefedik eltekintve az $\frac{1}{3n(9n)^k}$ szélességű hézagoktól.

3.0.2. Lemma. Legyen i_1, \dots, i_n , $q = 1, \dots, 2n+1$ és $k \geq 1$ konstans. Ekkor az $S_{k,i_1 \dots i_n}^q$ kockák lefedik az egész \mathcal{E}^n -nel jelölt n -dimenziós egységkockát, így minden \mathcal{E}^n -hez tartozó pont legalább $n+1$ -szer van fedve.

Tehát rögzített q mellett az $S_{k,i_1 \dots i_n}^q$ -k között $1/3n(9n)^k$ szélességű hézagok vannak, de q futása mellett ezek a hézagok eltűnnek, tehát csak így fedik le teljesen az \mathbb{R}^n -beli egységkockát. A következő lemma k -ra történő teljes indukcióval látható be.

3.0.3. Lemma. A konstans $\lambda_{k,i}^{pq}$ és ε_k (ahol p -re, q -ra, k -ra és i -re a fenti tulajdonságok érvényesek) megválaszthatók úgy, hogy a következők teljesüljenek:

1. $\lambda_{k,i}^{pq} < \lambda_{k,i+1}^{pq} \leq \lambda_{k,i}^{pq} + \frac{1}{2^k}$, tehát $\lambda_{k,i}^{pq}$ -k i futása mellett egy monoton növekvő sorozatot alkotnak, melynek differenciája kisebb mint $1/2^k$;

2. $\lambda_{k,i}^{pq} \leq \lambda_{k,i'}^{pq} \leq \lambda_{k,i}^{pq} + \varepsilon_k - \varepsilon_{k+1}$ ha az $A_{k,i}^q$ és $A_{k+1,i'}^q$ zárt intervallumok nem metszik egymást;
3. rögzített k -ra és q -ra a zárt $\Delta_{k,i_1 \dots i_n}^q = [\sum_{p=1}^n \lambda_{k,i_p}^{pq}; \sum_{p=1}^n \lambda_{k,i_p}^{pq} + n\varepsilon_k]$ intervallumok páronként diszjunktak;
4. $\varepsilon_k \leq \frac{1}{2^k}$.

Itt 1) és 3) implikálja 4)-et, majd ezeket felhasználva kaphatjuk az alábbi állítást:

3.0.4. Lemma. Legyen p és q rögzített, $k = 1, 2, \dots$ és $i = 1, \dots, m_k$. Ekkor a

$$\lambda_{k,i}^{pq} \leq \psi^{pq}(x) \leq \lambda_{k,i}^{pq} + \varepsilon_k$$

feltétel által meghatározott ψ^{pq} függvény létezik és egyértelmű, továbbá folytonos \mathcal{E}^1 -en $x \in A_{k,i}^q$ esetén.

A ψ^{pq} függvények fenti tulajdonságait felhasználjuk a fő tétel bizonyításánál.

Bizonyítás. Konstrukciójukból adódóan a ψ^{pq} függvények monoton növekedőek. Az előző lemmából és a 3) tulajdonságból következik, hogy

$$\sum_p \psi^{pq}(x_p) \in \Delta_{k,i_1 \dots i_n}^q$$

ahol $(x_1, \dots, x_n) \in S_{k,i_1 \dots i_n}^q$.

A ψ^{pq} függvények létezésének tényével valamint a $\lambda_{k,i}^{pq}$ és ε_i fenti tulajdonságaival a birtokunkban, most a $\chi^q(y)$ függvényeket határozzuk meg az alábbi határértékként:

$$\chi^q = \lim_{r \rightarrow \infty} \chi_r^q.$$

Itt χ_r^q -t r -re és k_r -re egyidejűleg történő teljes indukcióval fogjuk meghatározni, ahol k_r természetes szám. Bevezetjük a következő jelöléseket:

$$f_r(x_1, \dots, x_n) = \sum_q \chi_r^q \left[\sum_p \psi^{pq}(x_p) \right], \quad (3.2)$$

$$M_r = \sup_{\mathcal{E}^n} |f - f_r| \quad (3.3)$$

(ahol M_r az r . hibát jelöli). Nyilvánvaló, hogy $r = 0$ esetén $\chi_0^q = 0$, $f_0 = 0$, $k_r = 1$ és $M_0 = \sup_{\mathcal{E}^n} |f|$. Tegyük fel, hogy χ_{r-1}^q -et és k_{r-1} -et már meghatároztuk, így f_{r-1} is determinálva van \mathcal{E}^n -en. Mivel $S_{k,i_1 \dots i_n}^q$ átmérője nullához tart $k \rightarrow \infty$ esetén, ezért k_r -et megtudjuk választani úgy, hogy az $|f - f_{r-1}|$ különbség oszcillációja ne haladja meg $(\frac{1}{2^{n+2}})M_{r-1}$ -et egyik $S_{k_r,i_1 \dots i_n}^q$ -en sem. Legyen $\xi_{k,i}^q$ egy, a megfelelő $A_{k,i}^q$ -hez tartozó tetszőleges pont. Azt állítjuk, hogy a $\Delta_{k,i_1 \dots i_n}^q$ zárt intervallumon belül

$$\chi_r^q(y) = \chi_{r-1}^q(y) + \frac{1}{n+1} \left[f(\xi_{k,i_1}^q, \dots, \xi_{k,i_n}^q) - f_{r-1}(\xi_{k,i_1}^q, \dots, \xi_{k,i_n}^q) \right] \quad (3.4)$$

teljesül. Az ily módon meghatározott χ_r^q kielégíti a

$$|\chi_r^q(y) - \chi_{r-1}^q(y)| \leq \frac{1}{n+1} M_{r-1} \quad (3.5)$$

egyenlőtlenséget. A $\Delta_{k,i_1\dots i_n}^q$ intervallumon kívül χ_r^q -t tetszőlegesen definiálhatjuk, megőrizve a folytonosságot és az (3.5) tulajdonságot.

Most becsüljük meg $|f - f_r|$ különbséget egy tetszőleges (x_1, \dots, x_n) pontban, amely eleme \mathcal{E}^n -nek. Könnyen látható, hogy

$$\begin{aligned} f(x_1, \dots, x_n) - f_r(x_1, \dots, x_n) &= f(x_1, \dots, x_n) - f_{r-1}(x_1, \dots, x_n) - \\ &\quad - \sum_q \left[\chi_r^q \left(\sum_p \psi^{pq}(x_p) \right) - \chi_{r-1}^q \left(\sum_p \psi^{pq}(x_p) \right) \right]. \end{aligned} \quad (3.6)$$

A (3.6) egyenlőségben lévő \sum_q összeget felbontjuk két összegre, ahol az első A jelű összeg arra az $n+1$ darab q értékre terjed ki, amelyre az (x_1, \dots, x_n) pont csak egy $S_{k,i_1\dots i_n}^q$ kockának eleme, a második B jelű összeg pedig a fennmaradó n darab q -ra. Az (3.4) összefüggés miatt, az A jelű összeg minden tagjára

$$\begin{aligned} &\chi_r^q \left(\sum_p \psi^{pq}(x_p) \right) - \chi_{r-1}^q \left(\sum_p \psi^{pq}(x_p) \right) = \\ &= \frac{1}{n+1} \left(f(\xi_{k,i_1}^q, \dots, \xi_{k,i_n}^q) - f_{r-1}(\xi_{k,i_1}^q, \dots, \xi_{k,i_n}^q) \right) = \\ &= \frac{1}{n+1} \left(f(x_1, \dots, x_n) - f_{r-1}(x_1, \dots, x_n) \right) + \frac{\omega^q}{n+1} \end{aligned} \quad (3.7)$$

teljesül, ahol

$$|\omega^q| \leq \frac{1}{2n+2} M_{r-1}. \quad (3.8)$$

A B jelű összegben lévő tagok becsléséhez a (3.4), (3.5), (3.7) és a (3.8) összefüggéseket használjuk fel, így azt kaphatjuk, hogy:

$$\begin{aligned} |f - f_r| &= \left| \frac{1}{n+1} \sum_A \omega^q + \sum_B (\chi_r^q - \chi_{r-1}^q) \right| \leq \\ &\leq \frac{1}{2n+2} M_{r-1} + \frac{n}{n+1} = \frac{2n+1}{2n+2} M_{r-1} \end{aligned} \quad (3.9)$$

adódik. Az előző (3.9) egyenlőtlenség miatt minden $(x_1, \dots, x_n) \in E^n$ pontra

$$M_r \leq \frac{2n+1}{2n+2} M_{r-1}, \quad M_r \leq \left(\frac{2n+1}{2n+2} \right)^r M_0 \quad (3.10)$$

teljesül. Az előbbi (3.10) és az (3.5) egyenlőtlenségekből következik, hogy a

$|\chi_r^q - \chi_{r-1}^q|$ különbség értéke nem haladja meg a

$$\sum_r \frac{1}{n+1} M_{r-1}$$

abszolút konvergens sor megfelelő tagjának értékét és emiatt a χ_r^q függvények egyenletesen tartanak a folytonos χ^q határfüggvényhez, midőn $r \rightarrow \infty$. \square

4. fejezet

A szuperpozíciós tételben szereplő ψ függvény előállítása D. A. Sprecher nyomán

Kolmogorov 1957-ben megjelent eredményét először David A. Sprecher frissítette, amikor megmutatta, hogy a formulában szereplő $\psi^{p,q}$ függvény $\lambda_p \psi^q(x)$ alakban is felírható, ahol λ_p -k olyan konstansok, amikre $0 < \lambda_p \leq 1$ teljesül. Majd 1993-ban azt is belátta, hogy a tételben szereplő $\psi^{p,q}$ folytonos függvények megválaszthatók úgy is, hogy csak a változók számától, tehát n -től függjenek. Mi csak az utóbbi eredményt fogjuk bemutatni, ehhez bevezetjük a következő jelöléseket:

$$\mathcal{D}_0 = \left[0; \frac{1}{5!}\right],$$
$$\mathcal{D}_1 = \left[0; 1 + \frac{1}{5!}\right].$$

4.0.5. Definíció. Azt mondjuk, hogy az $\{\lambda_k\}$ sorozat együttesen független, ha $\sum t_p \lambda_p = 0$ akkor és csak akkor teljesül, ha $\sum |t_p| = 0$.

4.0.6. Tétel. Legyen $\{\lambda_k\}$ pozitív, együttesen független számok sorozata. Létezik egy folytonos, monoton növekvő $\psi : D_1 \rightarrow D_1$ függvény úgy, hogy a következők teljesülnek: minden valós értékű folytonos $f : \mathcal{E}^n \rightarrow R$ függvényhez, ahol $n \geq 2$, léteznek χ_q folytonos függvények, hogy

$$f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \chi_q \left(\sum_{p=1}^n \lambda_p \psi(x_p + qa_n) \right) \quad (4.1)$$

teljesül adott konstans a_n -re.

Lássuk tehát a ψ függvény egy a Kolmogorovétól eltérő konstrukcióját! A (4.1) összefüggésnek megfelelően ψ -t minden $n \geq 2$ -re a $[0; 1 + 2na_n]$ intervallumokon definiáljuk. A tételt a speciális $a_n = \sum_{r=2n+2}^{\infty} \frac{1}{r!}$ konstansra fogjuk belátni. A ψ függvényt először a

\mathcal{D}_0 intervallumon konstruáljuk meg, majd a következő összefüggéssel kiterjesztjük a \mathcal{D}_1 intervallumra:

$$\psi(x) = \psi\left(x - \frac{i}{5!}\right) + \frac{i}{5!}, \quad (4.2)$$

ahol $x \in [\frac{i}{5!}; \frac{i+1}{5!}]$ és $i = 0, 1, 2, \dots, 5!$. Legyen először $k = 5$ és $\{\rho_k\}$ olyan racionális számok sorozata, melyre teljesülnek a következők:

$$\rho_5 < \frac{1}{5!}, \quad 0 < \rho_{k+1} < \frac{\rho_k}{k}, \quad (4.3)$$

ahol $k \geq 5$. Behelyettesítve ψ -be $k = 5$ -re $\psi(0) = 0$ -át és $\psi(\frac{1}{5!}) = \frac{1}{5!}$ -t kapunk.

Most nézzük $k = 6$ -ra: legyen $j = 6i + t$, ahol $t = 0, 1, 2, \dots, 5$, ha $i = 0$ és $t = 0$, ha $i = 1$. Ekkor

$$\psi\left(\frac{j}{6!}\right) = \psi\left(\frac{i}{5!} + \frac{t}{6!}\right) = \begin{cases} \psi\left(\frac{i}{5!}\right) + \frac{t}{5}\rho_5 & \text{ha } t = 0, 1, 2, 3, 4, \\ \frac{1}{2}\left(\psi\left(\frac{i}{6!}\right) + \psi\left(\frac{i+1}{6!}\right)\right) & \text{ha } t = 5. \end{cases} \quad (4.4)$$

Ez a következő értékeket adja $\psi(\frac{j}{6!})$ -ra:

$$0, \quad \frac{1}{5}\rho_5, \quad \frac{2}{5}\rho_5, \quad \frac{3}{5}\rho_5, \quad \frac{4}{5}\rho_5, \quad \frac{1}{2 \cdot 5!}, \quad \frac{1}{5!}.$$

Tegyük fel, hogy $\psi(\frac{i}{(k-1)!})$ már ismert. Legyen $j = ki + t$, ahol $t = 0, 1, 2, \dots, k-1$, ha $i = 0, 1, 2, \dots, \frac{k!-1}{5}$; és $t = 0$, ha $i = \frac{k!}{5!}$. Ekkor az iterációs lépést az alábbi módon definiáljuk:

$$\begin{aligned} \psi\left(\frac{j}{k!}\right) &= \psi\left(\frac{i}{(k-1)!} + \frac{t}{k!}\right) = \\ &= \begin{cases} \psi\left(\frac{i}{(k-1)!}\right) + \frac{t}{k-1}\rho_{k-1} & \text{ha } t = 0, 1, \dots, k-2, \\ \frac{1}{2}\left(\psi\left(\frac{i}{(k-1)!}\right) + \psi\left(\frac{i+1}{(k-1)!}\right)\right) & \text{ha } t = k-1. \end{cases} \end{aligned} \quad (4.5)$$

A (4.2) valamint a (4.5) összefüggések alapján felírható a

$$\left|\psi\left(\frac{i+1}{k!}\right) - \psi\left(\frac{i}{k!}\right)\right| \leq \frac{1}{2^{k-5}5!}, \quad (4.6)$$

becslés, ahol $0 \leq i \leq \frac{k!-1}{5!}$ és $k \geq 5$. Ez az induktív eljárás az $[0; \frac{1}{5!}]$ intervallum pontjainak egy mindenhol sűrű halmazán definiálja a monoton növekedő ψ -t. Ezért az előző (4.6) becslés miatt $\psi(x)$ az egyetlen olyan folytonos, monoton növekvő függvény a \mathcal{D}_0 értelmezési tartományon, melynek képe tartalmazza a $(\frac{i}{k!}, \psi(\frac{i}{k!}))$ pontot.

Most lássuk a ψ függvény egy másik fontos tulajdonságát, amire szükségünk lesz a tétel bizonyításában:

4.0.7. Lemma. *Legyenek*

$$\delta_k = \frac{1}{k!} - \sum_{r=k+1}^{\infty} \frac{1}{r!}, \quad (4.7)$$

és

$$\nu_k = \sum_{r=k}^{\infty} \frac{r-1}{r} \rho_r. \quad (4.8)$$

Ekkor

$$\psi\left(\frac{i}{k!} + \delta_k\right) = \psi\left(\frac{i}{k!}\right) + \nu_k. \quad (4.9)$$

Bizonyítás. Egy tetszőleges rögzített $\frac{i}{k!}$ értékre és $N \geq k$ -ra tekintsük a

$$\xi_N = \frac{i}{k!} + \sum_{r=k+1}^N \frac{r-2}{r!} \quad (4.10)$$

pontot. A (4.5) iterációnak megfelelően a $\psi(\xi_N)$ értéknek megfelelő pont:

$$\begin{aligned} \psi(\xi_N) &= \psi\left(\frac{i}{k!}\right) + \sum_{r=k+1}^N \frac{r-2}{r-1} \rho_{r-1} \\ &= \psi\left(\frac{i}{k!}\right) + \sum_{r=k}^{N-1} \frac{r-1}{r} \rho_r \end{aligned} \quad (4.11)$$

Egy egyszerű számítás után azt kapjuk, hogy

$$\sum_{r=k+1}^N \frac{r-2}{r!} = \frac{1}{k!} - \sum_{r=k+1}^N \frac{1}{r!} - \frac{1}{N!}$$

és mivel mindegyik sorozat a fenti egyenlőségekben abszolút konvergens, azt kapjuk, hogy

$$\lim_{N \rightarrow \infty} \sum_{r=k+1}^N \frac{r-2}{r!} = \frac{1}{k!} - \lim_{N \rightarrow \infty} \left(\sum_{r=k+1}^N \frac{1}{r!} - \frac{1}{N!} \right) = \delta_k.$$

Így a (4.10) összefüggés azt adja, hogy

$$\lim_{N \rightarrow \infty} \xi_N = \frac{i}{k!} + \delta_k,$$

a (4.11) összefüggés pedig azt, hogy

$$\lim_{N \rightarrow \infty} \psi(\xi_N) = \psi\left(\frac{i}{k!}\right) + \lim_{N \rightarrow \infty} \sum_{r=k}^{N-1} \frac{r-1}{r} \rho_r = \psi\left(\frac{i}{k!}\right) + \nu_k.$$

Ebből pedig következik a lemma igazolása. \square

Minden pozitív $k \geq 5$ -re tekintsük az

$$E_k(i) = \left[\frac{i}{k!}; \frac{i}{k!} + \delta_k \right] \quad (4.12)$$

összefüggés által meghatározott zárt intervallumok rendszerét. Minden k értékre ezeket az intervallumokat egy-egy

$$e_k = \frac{1}{k!} - \delta_k = \sum_{r=k+1}^{\infty} \frac{1}{r!} \quad (4.13)$$

átmérőjű hézag választja el egymástól. Egyszerű számolással megmutatható, hogy ezek az e_k értékek az alábbi módon viszonyulnak az $E_k(i)$ intervallumok átmérőjéhez:

$$\delta_k > (k-1)e_k. \quad (4.14)$$

Ezeket a h zagokat lesz m tva az $E_k(i)$ intervallumok lefedik az eg sz \mathcal{D}_1 -et. K nny  be-
l tni, hogy

$$\begin{aligned} E_k(j) \subset E_{k-1}(i) \quad &\text{akkor  s csak akkor, ha} \\ j = ki + t, \quad &\text{ahol } t = 0, 1, 2, \dots, k-2. \end{aligned} \quad (4.15)$$

Tov bb , ha j  s i k z tt olyan kapcsolat van, amelyet az el z  (4.15)  sszef gg sben l tha-
tunk, akkor az $E_k(j)$  s az $E_{k-1}(i)$ intervallumok kezd pontjai $t = 0$ esetén, a v gpontjaik
pedig $t = k-2$ esetén esnek egybe. Ha pedig $t = k-1$, akkor a $E_k(j)$ intervallum az
 $E_{k-1}(i)$  s $E_{k-1}(i+1)$ intervallumok k z tti h zagba esik. Form lisan fel rhatjuk, hogy
az (4.12)-ben megjel lt $E_k(i)$ intervallumok  s azok ψ szerinti k pe k z tt a k vetke z 
kapcsolat  ll fenn:

$$\psi\left[\frac{i}{k!}; \frac{i}{k!} + \delta_k\right] = \left[\psi\left(\frac{i}{k!}\right); \psi\left(\frac{i}{k!}\right) + \nu_k\right] \quad (4.16)$$

A ψ f ggv ny monotonit sa miatt az intervallumok k pei p ronk nt diszjunktak lesznek
minden $k \geq 5$ -re.

A (4.0.6) t tel bizony t s hoz viszont sokkal szigor bb felt tetelt kell a $\{\rho_k\}$ racion lis
sz mokb l  ll  sorozatra kivetni, mint amit a (4.3)  sszef gg sben l thattunk:

4.0.8. Lemma. *Legyen $\{\lambda_k\}$ egy ttesen f ggetlen sz mok egy sorozata, ahol $\lambda_k > 0$. Min-
den $k \geq 5$ -re legyen*

$$\mu_k \sigma_k = \min_{i_p, j_p \in I_k} \left| \sum_{p=1}^{k-3} \lambda_p \left(\psi\left(\frac{i_p}{k!}\right) - \psi\left(\frac{j_p}{k!}\right) \right) \right|, \quad \sum_{p=1}^{k-3} |i_p - j_p| \neq 0 \quad (4.17)$$

ahol

$$\sigma_k = 1 + \sum_{p=1}^{k-3} \lambda_p. \quad (4.18)$$

A (4.3) beli racion lis ρ_k -k megv laszthat k  gy, hogy $\nu_k < \mu_k$ teljes lj n.

Bizony t s. Mivel a $\psi(i/k!)$ sz mok racion lisak, ez rt $\mu_k \neq 0$ minden k -ra,  s egy r-
telm , hogy az (4.0.7) lemm ban le rt konstrukci  akkor is  rv nyben marad, ha az ott
szerepl  ρ_k -kat fel l rjuk olyan $\tilde{\rho}_k$ -kal, melyekre teljes l a $0 < \tilde{\rho}_k < \rho_k$  sszef gg s. Mivel
 $\rho_k < \rho_{k-1}/(k-1)$ a (4.3)  rtelm ben, ez rt elegend  azt megmutatni, hogy a $\tilde{\rho}_k$ sz mok
megv laszthat k  gy, hogy a $0 < \tilde{\rho}_k < \min\{\mu_k, \tilde{\rho}_{k-1}/(k-1)\}$ egyenl tlens get kiel g ts k.
Ez megoldhat  egy boot-strap elj r ssal, a k vetke z  m don:

a $\psi(i/5!) = i/5!$  rt kek seg ts g vel, ahol $i \in I_5$ defini ljuk a k vetke z t:

$$\mu_5 \sigma_5 = \min_{i_p, j_p \in I_5} \left| \sum_{p=1}^2 \lambda_p \left(\psi\left(\frac{i_p}{5!}\right) - \psi\left(\frac{j_p}{5!}\right) \right) \right|, \quad \sum_{p=1}^2 |i_p - j_p| \neq 0$$

ahol $\sigma_5 = 1 + \lambda_1 + \lambda_2$, ezut n v lasztunk egy racion lis sz mot, amelyre teljes l, hogy
 $0 < \tilde{\rho}_5 < \mu_5$. Ezt a $\tilde{\rho}_5$ -t az (4.4)  sszef gg sbe behelyettes tve meghat rozzuk a k vetke z t:

$$\mu_6 \sigma_6 = \min_{i_p, j_p \in I_6} \left| \sum_{p=1}^3 \lambda_p \left(\psi\left(\frac{i_p}{6!}\right) - \psi\left(\frac{j_p}{6!}\right) \right) \right|, \quad \sum_{p=1}^3 |i_p - j_p| \neq 0,$$

ahol $\sigma_6 = 1 + \lambda_1 + \lambda_2 + \lambda_3$ majd megint választunk egy racionális számot, amelyre $0 < \tilde{\rho}_6 < \min\{\mu_6, \tilde{\rho}_5/5!\}$ teljesül. Tegyük fel, hogy $\psi(i/(k-1)!)$ -et és ρ_{k-1} -et már ismerjük. Ezekkel az értékekkel meghatározzuk $\psi(i/k!)$ értékét is (4.5) alapján, majd a (4.17)-ben szereplő $\mu_k \sigma_k$ értéket is $i \in I_k$ -ra. Ezt követően újra kiválasztunk egy racionális számot, mely kielégíti a $0 < \tilde{\rho}_k < \min\{\mu_k, \tilde{\rho}_{k-1}/(k-1)\}$ egyenlőtlenségeket. \square

Most, hogy meghatároztuk a (4.0.7) lemmában szereplő ρ_k -t kielégítve a (4.0.8) lemma feltételeit, végeztünk a ψ függvény konstrukciójával, így rátérhetünk az (4.0.6) tétel bizonyítására.

Bizonyítás. Az $E_k(i)$ és az $[\psi(i/k!), \psi(i/k!) + \nu_k]$ intervallumok közötti kölcsönösen egyértelmű megfeleltetés miatt az $E_k(i)$ intervallumok minden nem üres metszetének, az $[\psi(i/k!), \psi(i/k!) + \nu_k]$ intervallumok csak egyetlen nem üres metszete felel meg.

Legyen adott egy $n \geq 2$ változó. Minden $q = 0, 1, 2, \dots, 2n$ változóra meghatározzuk zárt intervallumok egy alábbi rendszerét:

$$E_k^q(i) = \left[\frac{i}{k!} - qe_{2n+1}; \frac{i}{k!} + \delta_k - qe_{2n+1} \right], \quad (4.19)$$

ahol $i \in I_k$ és $k \geq 5$. Ezek az intervallumok az $E_k(i)$ -k $-qe_{2n+1}$ -el való eltolásai, továbbá k és q minden értékére léteznek olyan i' és i'' változók, melyekre $0 \in E_k^q(i')$ és $1 + (1/5!) \in E_k^q(i'')$ teljesülnek. Minthogy a ψ függvény konstrukciójában az i index csak k -tól függ, egyértelmű, hogy most q -tól is függni fog. Az indexek változásai könnyen kiszámíthatóak, mi azonban azt fogjuk feltenni, hogy mindig megfelelően vannak kiválasztva. Figyelembe véve a fenti (4.19) definíciót, most részletezzük a (4.1) formulában szereplő a_n konstans alakját, melynek értéke:

$$a_n = e_{2n+1} = \sum_{r=2n+2}^{\infty} \frac{1}{r!}. \quad (4.20)$$

4.0.9. Lemma. *Legyen adott egy $n \geq 2$ változó. Ekkor minden $k \geq 2n + 1$ és $q = 0, 1, 2, \dots, 2n$ értékre*

$$\text{diam}[E_k^q(i)] \rightarrow 0 \quad k \rightarrow \infty \quad \text{esetén és} \quad (4.21)$$

$$E_k^q(i) \cap E_k^q(j) = 0, \quad \text{ha } i \neq j. \quad (4.22)$$

Minden $x \in D_1$ pontra legalább $2n$ darab q érték létezik amelyre $x \in E_k^q(i)$ fennáll.

Bizonyítás. Ez a lemma egyértelmű következménye az $E_k(i)$ -k tulajdonságainak, csak az utolsó sorban lévő állítás szorul egy kis magyarázatra. Megjegyezzük, hogy az $E_k^q(i)$ intervallumokat a

$$G_k^q(i) = \left(\frac{i}{k!} + \delta_k - qe_{2n+1}; \frac{i+1}{k!} - qe_{2n+1} \right) \quad (4.23)$$

nyílt hézagok választják el egymástól. Megvizsgálva a $G_k^q(i')$ és a $G_k^{q+1}(i'')$ intervallumokat, egy egyszerű számítás után azt vehetjük észre, hogy a $G_k^q(i')$ kezdőpontja egybeesik $G_k^{q+1}(i'')$ határpontjával, ha az i' és i'' indexekre teljesül az $i' - i'' = k!(e_k - e_{2n+1}) - 1$ egyenlőség. Így a (4.13) és a (4.14) formulák miatt a $G_k^q(i)$ -k nem metszik egymást $q =$

$0, 1, 2, \dots, 2n$ és rögzített k esetén. Ebből az következik, hogy minden $x \in D_1$ pont legfeljebb csak egy $G_k^q(i)$ -hez tartozhat és a lemma végén kimondott állítás pedig következik abból a tényből, hogy minden x -nek tartoznia kell egy $E_k^q(i)$ vagy egy $G_k^q(i)$ intervallumhoz minden q érték esetén. \square

4.0.10. Lemma. *Legyen adott egy $n \geq 2$ változó és legyen $k \geq 2n+1$ és $q = 0, 1, 2, \dots, 2n$. Minden $1 \leq p \leq n$ -re legyenek a $\{E_k^q(i_p)\}$ intervallumcsaládok az \mathcal{E}^n egységkocka p -edik koordinátatengelyén. Az $E_k^q(i_p)$ intervallumokhoz hasonlóan megfigyeljük a következő*

$$T_k^q(i_1, \dots, i_n) = \left(\sum_{p=1}^n \lambda_p \psi\left(\frac{i_p}{k!} + qe_{2n+1}\right); \sum_{p=1}^n \lambda_p \left(\psi\left(\frac{i_p}{k!} + qe_{2n+1}\right) + \nu_k \right) \right) \quad (4.24)$$

intervallumokat. Ekkor

$$\text{diam}[T_k^q(i_1, \dots, i_n)] \rightarrow 0 \quad k \rightarrow \infty \quad \text{esetén és} \quad (4.25)$$

$$\begin{aligned} T_k^q(i_1, \dots, i_n) \cap T_k^q(j_1, \dots, j_n) &= \emptyset \quad \text{ha} \\ (i_1, \dots, i_n) &\neq (j_1, \dots, j_n). \end{aligned} \quad (4.26)$$

Legyen továbbá

$$\xi_q = \sum_{p=1}^n \lambda_p \psi(x_p + qe_{2n+1}). \quad (4.27)$$

Ekkor minden $(x_1, \dots, x_n) \in \mathcal{E}^n$ pontra legalább $n+1$ darab q érték létezik, amellyel $\xi_q \in T_k^q(i_1, \dots, i_n)$ teljesül.

Bizonyítás. A (4.25)-ös állítás a (4.24)-ös definícióból következik, míg a (4.26)-as állítás érvényessége a következőkből látható: az $E_k(i)$ és $E_k^q(i)$ intervallumok az $y^q = x + qe_{2n+1}$ leképezés révén állnak kapcsolatban egymással, így $x \in E_k(i)$ akkor és csak akkor teljesül, ha $y^q \in E_k^q(i)$. Ezért a μ_k , amit a (4.17)-es formulában láthattunk marad változatlan, mikor $\psi(x)$ -et $\psi(x + qe_{2n+1})$ -re és $E_k(i)$ -t $E_k^q(i)$ -re cseréljük. A (4.0.8) lemma értelmében, alkalmazva a (4.24) egyenlőséget és figyelembe véve azt is, hogy $\nu_k < \mu_k$, azt látjuk, hogy

$$\begin{aligned} \text{diam}[T_k^0(i_1, \dots, i_n)] &= \nu_k \sum_{p=1}^{k-3} \lambda_p < \mu_k \sum_{p=1}^{k-3} \lambda_p \\ &= \frac{1}{\sigma_k} \left(\sum_{p=1}^{k-3} \lambda_p \right) \times \min_{i_p, j_p \in I_k} \left| \sum_{p=1}^{k-3} \lambda_p \left(\psi\left(\frac{i_p}{k!}\right) - \psi\left(\frac{j_p}{k!}\right) \right) \right| < \\ &< \min_{i_p, j_p \in I_k} \left| \sum_{p=1}^{k-3} \lambda_p \left(\psi\left(\frac{i_p}{k!}\right) - \psi\left(\frac{j_p}{k!}\right) \right) \right|. \end{aligned}$$

A lemma utolsó állítása egyértelmű következménye a (4.0.9) lemmának: legyen adott egy k változó és egy $(x_1, \dots, x_n) \in \mathcal{E}^n$ pont. A (4.22) állításnak megfelelően legfeljebb egy olyan q érték létezik, hogy egyik p -re sem teljesül, hogy $x_p \in E_k^q(i_p)$. Ezért legfeljebb n darab q érték létezik amelyre $\xi_q \notin T_k^q(i_1, \dots, i_n)$, ahol ξ_q a (4.27)-es összefüggésben látható módon kapható (x_1, \dots, x_n) -ből. Így $\xi_q \in T_k^q(i_1, \dots, i_n)$ legalább $n+1$ darab q értékre, mi pedig ezt akartuk megmutatni. \square

Ezzel pedig a szakasz elején tárgyalt tételt beláttuk. \square

5. fejezet

A szuperpozíciós tétel és a neurális hálók kapcsolata

Robert Hecht-Nielsen volt az első, aki elkezdte használni ezeket az eredményeket a neurális hálók területén, az ő nevéhez köthető az alábbi tétel is 1987-ből.

5.0.11. Tétel. *Minden folytonos $f : \mathcal{E}^n \rightarrow \mathbb{R}$ függvény, ahol $n \geq 2$ implementálható egy háromrétegű feedforward neurális háló által, melynek n neuronja van az első input rétegen, $2n + 1$ a második rétegen és szintén $2n + 1$ darab a harmadik, kimeneti rétegen. Az első két réteg elemei függetlenek n -től és csak a harmadik réteg függ az f függvénytől.*

A szuperpozíciós tétel formulája és egy neurális háló által realizált függvény hasonlóságából Hecht-Nielsen arra következtetett, hogy minden az n -dimenziós \mathbb{R}^n -beli egységkockán értelmezett függvény pontosan reprezentálható egy bizonyos neurális háló segítségével. Vera Kurkova pedig megmutatta, hogy az eredeti konstrukció megváltoztatásával egy lehetséges módot kapunk a függvények közelítésére. Ennélfogva minden folytonos függvény tetszőleges pontossággal közelíthető egy perceptronnal, melynek két rejtett rétegen nagyobb számú elem van és egy standard szigmoid aktivációs függvénnyel rendelkezik. Ahhoz, hogy ennek módját megmutassuk be kell vezetnünk néhány definíciót.

5.0.12. Definíció. *Sigmoid függvénynek nevezünk minden olyan $\sigma : \mathbb{R} \rightarrow \mathcal{E}^n$ függvényt, melyre teljesülnek az alábbi feltételek:*

$$\lim_{t \rightarrow -\infty} \sigma(t) = 0 \quad \lim_{t \rightarrow \infty} \sigma(t) = 1.$$

5.0.13. Definíció. *Az összes*

$$f(x) = \sum_{i=1}^k a_i \sigma(b_i x + c_i)$$

alakú f függvényt tartalmazó osztályt σ típusú lépcsős függvények halmazának nevezzük és $\mathcal{S}(\sigma)$ -val jelöljük.

5.0.14. Definíció. *$\mathcal{C}(\mathcal{E}^n)$ -nel jelöljük az \mathcal{E}^n -en folytonos függvények halmazát.*

Ezen függvényeket akarjuk majd közelíteni.

5.0.15. Definíció. Az $f : \mathcal{E}^n \rightarrow \mathbb{R}$ függvény folytonossági modulusának nevezzük a $\omega_f : (0; \infty) \rightarrow \mathbb{R}$ függvényt, ha

$$\omega_f(\delta) = \sup \left\{ |f(x_1, \dots, x_n) - f(y_1, \dots, y_n)|; \right. \\ \left. \text{ahol } (x_1, \dots, x_n) \text{ és } (y_1, \dots, y_n) \in \mathcal{E}^n \text{ valamint} \right. \\ \left. |x_p - y_p| < \delta \text{ minden } p = 0, 1, \dots, n \right\}.$$

V. Kurkova tehát mind a külső χ , mind a belső ψ függvény közelítésére mutatott egy-egy utat.

5.0.16. Tétel. Legyen $n \in \mathbb{N}$, $n \geq 2$, $\sigma : \mathbb{R} \rightarrow \mathcal{E}$ szigmoid függvény, $f \in \mathcal{C}(\mathcal{E}^n)$ és ε pozitív, valós szám. Ekkor létezik $k \in \mathbb{N}$ és $\chi_i, \psi_{pi} \in \mathcal{S}(\sigma)$ függvények, melyekre

$$\left| f(x_1, \dots, x_n) - \sum_{i=1}^k \chi_i \left(\sum_{p=1}^n \psi_{pi}(x_p) \right) \right| \leq \varepsilon \quad (5.1)$$

teljesül minden $(x_1, \dots, x_n) \in \mathcal{E}^n$ esetén.

5.0.17. Tétel. Legyen $n \in \mathbb{N}$ és $n \geq 2$, $\sigma : \mathbb{R} \rightarrow \mathcal{E}$ szigmoid függvény, $f \in \mathcal{C}(\mathcal{E}^n)$ és $\varepsilon \geq 0$ valós szám. Ekkor minden $m \in \mathbb{N}$ számra, amelyre $m \geq 2n + 1$ és $n/(m - n) + v < \varepsilon/\|f\|$ és $\omega_f(1/m) < v(m - n)/(2m - 3n)$ teljesül valós pozitív v esetén, f függvény ε pontossággal approximálható egy két rejtett rétegű perceptronnal, amely σ aktivációs függvénnel rendelkezik és amelynek az első rejtett rétegén $nm(m + 1)$ darab, a másodikon pedig $m^2(m + 1)^n$ darab neuron van.

Először felvázoljuk az algoritmus fő lépéseit:

- létrehozunk egy \mathcal{Q}^n racionális \mathbf{d}_k pontokból álló hálót, amely az egységkockát bontja fel,
- a ξ függvény konstrukciója a \mathbf{d}_k^q pontokban,
- θ szigmoid lépés,
- a külső χ_q^j függvény θ -n és az előző iterációs lépés e_r hibáján alapuló szerkesztése,
- az eredeti f függvény r -dik f_r közelítése és az r -dik iterációs lépésből adódó e_r hiba meghatározása.

Most lássuk mindezt részletesebben is. Legyen $m \geq 2n$ és $\gamma \geq m + 2$, ahol n a bemenő adatok dimenziója. Nézzük a racionális számokból álló

$$\mathcal{Q} = \left\{ d_k = \sum_{s=1}^k i_s \gamma^{-s}; \quad i_s \in \{0, 1, \dots, \gamma - 1\}, \quad k \in \mathbb{N} \right\}$$

halmazt. \mathcal{Q} elemeire úgy is tekinthetünk, mint egy n -dimenziós háló

$$\mathcal{Q}^n = \{\mathbf{d}_k = (d_{k1}, \dots, d_{kn}); \quad d_{kj} \in \mathcal{Q}, \quad j = 1, 2, \dots, n\} \quad (5.2)$$

koordinátáira. A $\mathbf{d}_k^q \in \mathcal{Q}^n$ koordinátákat, melyeket fel akarunk majd használni a ξ függvény $(\xi(\mathbf{x}_q) = \sum_{p=1}^n \lambda_p \psi(x_p + q\beta))$, az előző fejezetben, a (4.1) összefüggésben látható függvény, amit χ_q függvénybe helyettesítünk) definiálásánál $q = 0, 1, 2, \dots, m$ -re a

$$d_{kp}^q = d_{kp} + q \sum_{s=1}^k \gamma^{-s} \quad (5.3)$$

összefüggés határozza meg. Világos, hogy $d_{kp}^q \in \mathcal{Q}$ $p = 1, 2, \dots, n$ esetén.

Most rátérhetünk a belső $\psi : \mathcal{Q} \rightarrow \mathcal{E}$ függvényre.

$$\psi(d_k) = \sum_{s=1}^k \tilde{i}_s 2^{m_s} \gamma^{-\varrho(s-m_s)}, \quad \text{ahol} \quad (5.4)$$

$$\begin{aligned} \varrho(z) &= \frac{n^z - 1}{n - 1}, \\ m_s &= \langle i_s \rangle \left(1 + \sum_{l=1}^{s-1} [i_l] \cdots [i_{s-1}] \right) \quad \text{és} \\ \tilde{i}_s &= i_s - (\gamma - 2) \langle i_s \rangle. \end{aligned}$$

Legyen $[i_1] = \langle i_1 \rangle = 1$. Ha $s \geq 2$ akkor $[i_s]$ -t és $\langle i_s \rangle$ -t a következő módon definiáljuk:

$$\begin{aligned} [i_s] &= \begin{cases} 0 & \text{ha } i_s = 0, 1, 2, \dots, \gamma - 3 \\ 1 & \text{ha } i_s = \gamma - 2, \gamma - 1 \end{cases} \\ \langle i_s \rangle &= \begin{cases} 0 & \text{ha } i_s = 0, 1, 2, \dots, \gamma - 2 \\ 1 & \text{ha } i_s = \gamma - 1 \end{cases} \end{aligned}$$

Most rátérünk a külső χ függvényre. A χ_q függvényeket iteratív módon állítjuk elő a következő függvénysor határértékeként:

$$\chi_q(y_p) = \lim_{r \rightarrow \infty} \sum_{j=1}^r \chi_q^j(y_q).$$

Minden $\chi_q^j(y_q)$ függvényt e_{j-1} (a $j - 1$ -edik iterációs lépésből származó hiba) által határozzuk meg \mathcal{Q}^n pontjaiban:

$$\chi_q^j \circ \xi(\mathbf{x}_q) = \frac{1}{m+1} \sum_{\mathbf{d}_k^q} e_{j-1}(\mathbf{d}_k) \theta(\mathbf{d}_k^q; \xi(\mathbf{x}_q)), \quad (5.5)$$

ahol $q = 0, 1, \dots, m$, $j = 1, 2, \dots, r$ és $\mathbf{d}_k \in \mathcal{Q}$. A valós értékű $\theta(\mathbf{d}_k; \xi(\mathbf{x}_q))$ függvényt a \mathcal{Q}^n egy rögzített \mathbf{d}_k^q pontjára definiáljuk egy adott σ szigmoid függvény segítségével a következő módon:

$$\theta(\mathbf{d}_k^q; y_q) = \sigma(\gamma^{\beta(k+1)}(y_q - \xi(\mathbf{d}_k^q)) + 1) - \sigma(\gamma^{\beta(k+1)}(y_q - \xi(\mathbf{d}_k^q)) - (\gamma - 2)b_k), \quad (5.6)$$

ahol $y_q \in \mathbb{R}$. A fenti formulában b_k egy valós szám, melyet a következő összefüggés határoz meg:

$$b_k = \sum_{s=k+1}^{\infty} \gamma^{-\varrho(s)} \sum_{p=1}^n \lambda_p.$$

A $\xi(\mathbf{d}_k^q)$ függvényt pedig az alábbi egyenlőséggel definiáljuk:

$$\xi(\mathbf{d}_k^q) = \sum_{p=1}^n \lambda_p \psi(d_{kp}^q), \quad (5.7)$$

ahol $\psi(d_{kp}^q)$ a (5.4)-es összefüggésből származik, λ -t pedig a következők szerint adjuk meg:

$$\lambda_p = \sum_{s=1}^{\infty} \gamma^{-(p-1)\varrho(s)}, \quad (5.8)$$

itt $\lambda_1 = 1$ és $p > 1$.

Most nézzük az iterációs lépést részletesebben. Legyen f egy ismert folytonos függvény, az iteráció kezdetén $e_0 = f$. Az r -dik approximációs hibafüggvényt e_r -t így számoljuk $r = 1, 2, \dots$ esetén:

$$e_r(\mathbf{x}) = e_{r-1}(\mathbf{x}) - \sum_{q=0}^m \chi_q^r \circ \xi(\mathbf{x}_q), \quad (5.9)$$

ahol $\mathbf{x} \in \mathcal{E}^n$, $\mathbf{x}_q = (x_1 + q\beta, \dots, x_n + q\beta)$ és $\beta = \gamma/(\gamma - 1)$. Majd az r -dik f_r approximációs függvény

$$f_r(\mathbf{x}) = \sum_{j=1}^r \sum_{q=0}^m \chi_q^j \circ \xi(\mathbf{x}_q) \quad (5.10)$$

szerint adódik. Sprecher pedig megmutatta, hogy $r \rightarrow \infty$ esetén $f_r \rightarrow f$.

Még a legkisebb mennyiségű bemeneti adat esetén is nagyon időigényesek lehetnek a számítások. Ahhoz, hogy megkapjuk a megfelelő számítási eredményeket nagyon kell vigyáznunk arra, hogy kikerüljünk a lehetséges akadályokat. A d_k értékeket nem csak a saját értékükkel reprezentálhatjuk, hanem a saját i_s tagjaikkal, amik révén gyorsabban számíthatjuk ki a $\psi(d_k)$ értékeket. Néhány kifejezést, mint m_s -t és χ_q^j -t megvizsgáltak és átírták sokkal hatásosabb, vagy használhatóbb alakba.

5.1. A differenciálhatóság problémája

Több cikk is boncolgatja azt a kérdést, hogy a függvények neurális hálókkal való közelítéséhez mennyire volt fontos Kolmogorov eredménye. Az approximáció elmélet kritikus pontja ugyanis a közelítő függvény reprezentálásának módja. Mivelhogy minden reprezentáció felírható egy alkalmas neurális hálóval ezért a reprezentáció gyakorlatilag ekvivalens a neurális háló szerkezetének kiválasztásával.

A Kolmogorov tétel neurális hálókkal való értelmezése nagyon vonzó: egy függvény reprezentálása rögzített számú neuront igényel, melyek az input tér dimenziójával együtt növekednek. A probléma azonban a Kolmogorov formula belső függvényével van, mivel

ezek a függvények folytonosak, Vitushkin és Henkin belátták, hogy magasabb rendben nem differenciálhatónak kell lenniük.

5.1.1. Vitushkin tétel (1954). *Léteznek r -szer folytonosan differenciálható n változós függvények, ahol $r = 1, 2, \dots$ és $n \geq 2$, amelyek nem állíthatók elő r -szer folytonosan differenciálható m változós függvények szuperpozíciójaként, ahol $m < n$; továbbá léteznek olyan kétváltozós r -szer folytonosan differenciálható függvények, melyek nem állíthatók elő összegekkel és egyváltozós folytonosan differenciálható függvényekkel.*

Megjegyezzük, hogy Hilbert tizenharmadik problémáját és a (5.1.1) tételt is ugyanaz az intuíció alapozta meg: nem minden komplexebb függvény reprezentálható egyszerűen kevésbé összetett függvények átlagával. Hilbert problémájának "hibája" az, hogy rosszul definiálja a függvények komplexitását. A Kolmogorov tétel rámutat, hogy csupán a változók száma nem elegendő ahhoz, hogy megfelelően karakterizálja egy függvény bonyolultságát. Erre adott megoldást Vitushkin.

Legyen f egy r -szer folytonosan differenciálható függvény \mathcal{E}^n -en, melynek minten r -ed rendű parciális deriváltja Lipschitz folytonos $[0, 1]$ -en. Vitushkin meghatározott egy $\chi = (r + \alpha)/n$ értéket és megmutatta, hogy ez jól karakterizálja annak a mértékét, hogy egy adott függvény mennyire nem bonyolult. Itt α a Lipschitz folytonosság együtthatója. Ezt láthatjuk a következő tételben:

5.1.2. Vitushkin. *Léteznek olyan adott $\chi_0 = q_0/k_0$ karakterisztikájú függvények, amelyek nem állíthatók elő $\chi = q/k$ karakterisztikájú függvények szuperpozíciójaként, ahol $\chi > \chi_0$ és $q \geq 1$.*

Vera Kurkova szerint viszont igen is fontos, meghatározó mérföldkő volt Kolmogorov formulája a neurális hálók elméletében. Az

$$f(x_1, x_2, \dots, x_n) = \sum_{q=1}^{2n+1} \varphi_q \left[\sum_{p=1}^n \psi_{pq}(x_p) \right]$$

összefüggés könnyen összekapcsolható a perceptron típusú neurális hálókkal azzal a korlátozással, hogy az egyváltozós φ_q és ψ_{pq} függvények nem állnak közel azokhoz a függvényekhez, melyeket a neurális hálózatokkal kapcsolatos számításokban alkalmaznak.

V. Kurkova megmutatta, hogy a Kolmogorov formulában lévő egyenlőség egy approximációval való helyettesítésével eltüntethetjük a szakasz első felében említett problémákat, melyek a függvények simaságából fakadnak. A neurális hálók kontextusában minket csak a függvények approximációja érdekel. Az egyetlen probléma a tétel relevanciájára vonatkozóan csak az, hogy Kolmogorov konstrukciója megváltoztatható-e úgy, hogy minden egyváltozós függvény olyan kellően sima függvények sorozatának határértéke legyen, melyeket a perceptron típusú hálókból használunk. Ezek a függvények a $[0, 1]$ intervallum affin transzformációiból képzett kompozíciók lineáris kombinációi adott szigmoid függvényekkel. Az ilyen típusú függvényeket szigmoid típusú lépcsős függvényeknek nevezzük.

A φ_q és a ψ_{pq} függvények tökéletesen alkalmasak tetszőleges szigmoid lépcsős függvényeknek. A függvényeknek amelyeket a φ_p és ψ_{pq} függvények konstrukciójában használunk egyetlen igazán fontos tulajdonsága van: az, hogy végesen sok zárt intervallumon meghatározott értékeket vegyenek fel, ezeken kívül tetszőlegesek lehetnek, azzal a feltétellel, hogy kellően korlátosak. Az ilyen függvények pedig tetszőlegesen jól közelíthetők tetszőleges szigmoid típusú lépcsős függvényekkel.

6. fejezet

Függvényapproximáció szigmoid függvények segítségével

Ebben a fejezetben megmutatjuk, hogy a rögzített egyváltozós függvények kompozíciójának véges lineáris kombinációi egyenletesen tudnak közelíteni tetszőleges n -változós függvényeket, úgy hogy csak gyenge feltételeket vetünk ki az egyváltozós függvényre. Beszélünk a reprezentálhatóságról az egy rejtett réteggel rendelkező neurális hálók osztályában. Majd megmutatjuk azt is, hogy tetszőleges döntési régió tetszőlegesen jól közelíthető egy folytonos, egy rejtett réteggel rendelkező neurális háló segítségével. Végül kitérünk más lehetséges aktivációs függvények approximációs tulajdonságaira.

Tekintsük a következő összefüggést:

$$\sum_{j=1}^N \alpha_j \sigma(y_j^T x + \theta_j), \quad (6.1)$$

ahol $y_j \in \mathbb{R}^n$ és α_j, θ rögzített valós számok. Itt az egyváltozós σ függvény nagyban függ az adott alkalmazási területtől. Mivel mi most neurális hálózatokkal foglalkozunk, így természetesen adódik, hogy a σ függvényt itt is szigmoid függvénynek válasszuk.

Elsősorban azt szeretnénk megmutatni, hogy a (6.1) összefüggésben szereplő összeg sűrű a folytonos függvények terében, ha σ egy folytonos szigmoid függvény.

6.1. A $\mathcal{C}(\mathcal{E}^n)$ -beli sűrűség

Legyen tehát \mathcal{E}^n újra az \mathbb{R}^n -beli egységkocka, vagyis $[0; 1]^n$, $\mathcal{C}(\mathcal{E}^n)$ az ezen értelmezett folytonos függvények tere, $\|f\|$ pedig a szuprémum normája egy adott $\mathcal{C}(\mathcal{E}^n)$ belüli f függvénynek. A véges, előjeles, reguláris Borel-mértékek összességét \mathcal{E}^n -en $M(\mathcal{E}^n)$ jelöli. A következőkben azokra a feltételeket próbáljuk megmutatni, melyek mellett a

$$G(x) = \sum_{j=1}^N \alpha_j \sigma(y_j^T x + \theta_j)$$

összegek sűrűek lesznek $\mathcal{C}(\mathcal{E}^n)$ -en a szuprémum normában. Ehhez bevezetünk egy definíciót.

6.1.1. Definíció. Azt mondjuk, hogy σ diszkriminátor függvény, ha egy $\mu \in M(\mathcal{E}^n)$ mértékre

$$\int_{\mathcal{E}^n} \sigma(y^T x + \theta) d\mu(x) = 0$$

egyenlőségből következik, hogy $\mu = 0$ minden $y \in \mathbb{R}^n$ és $\theta \in \mathbb{R}$ esetén.

6.1.2. Tétel. Legyen σ egy folytonos diszkriminátor függvény. Ekkor a

$$G(x) = \sum_{j=1}^N \alpha_j \sigma(y_j^T x + \theta_j) \quad (6.2)$$

összefüggés által definiált véges összeg sűrű $\mathcal{C}(\mathcal{E}^n)$ -ben, vagyis tetszőleges adott $f \in \mathcal{C}(\mathcal{E}^n)$ -re és $\varepsilon > 0$ -ra létezik egy a fenti összefüggés szerint definiált összeg amelyre

$$|G(x) - f(x)| < \varepsilon$$

minden $x \in \mathcal{E}^n$ esetén.

Bizonyítás. Legyen $S \subset \mathcal{C}(\mathcal{E}^n)$ a (6.2) összefüggésben szereplő függvények halmaza. Világos, hogy S lineáris altere $\mathcal{C}(\mathcal{E}^n)$ -nek. Ekkor S lezártja melyet innentől R -nek nevezünk nem egyenlő az egész $\mathcal{C}(\mathcal{E}^n)$ -nel, ezért R valódi zárt altere $\mathcal{C}(\mathcal{E}^n)$ -nek. A Hahn-Banach tétel szerint, ekkor létezik egy korlátos lineáris funkcionál $\mathcal{C}(\mathcal{E}^n)$ -n, nevezzük L -nek, melyre teljesülnek a következő tulajdonságok: $L \neq 0$, de $L(R) = L(S) = 0$. A Riesz reprezentációs tétel miatt ez a funkcionál minden $h \in \mathcal{C}(\mathcal{E}^n)$

$$L(h) = \int_{\mathcal{E}^n} h(x) d\mu(x)$$

alakú, ahol $\mu \in M(\mathcal{E}^n)$. Mivel $\sigma(y^T x + \theta) \in R$ minden y -ra és θ -ra, ezért nekünk az kell, hogy

$$\int_{\mathcal{E}^n} \sigma(y^T x + \theta) d\mu(x) = 0$$

minden y -ra és θ -ra. Mivel feltettük, hogy σ diszkriminátor függvény ezért következik, hogy $\mu = 0$, ami ellentmond a feltevésünknek. Így S altérnek sűrűnek kell lennie $\mathcal{C}(\mathcal{E}^n)$ -ben \square

Ez a bizonyítás megmutatta, hogy a (6.2) alakú összegek sűrűek $\mathcal{C}(\mathcal{E}^n)$ -ben, ha σ egy folytonos diszkriminátor függvény. A következőkben megmutatjuk, hogy bármely folytonos szigmoid függvény diszkriminátor.

6.1.3. Lemma. Bármely korlátos, mérhető szigmoid függvény diszkriminátor, speciálisan bármely folytonos szigmoid függvény diszkriminátor.

Bizonyítás. Ahhoz, hogy ezt megmutassuk, először is megjegyezzük, hogy bármely x -re, y -ra, θ -ra és ϕ -re

$$\sigma(\lambda(y^T x + \theta) + \phi) \begin{cases} \rightarrow 1 & \text{ha } y^T x + \theta > 0 \quad \lambda \rightarrow +\infty \text{ esetén,} \\ \rightarrow 0 & \text{ha } y^T x + \theta < 0 \quad \lambda \rightarrow -\infty \text{ esetén,} \\ = \sigma(\phi) & \text{ha } y^T x + \theta = 0 \quad \text{minden } \lambda\text{-ra.} \end{cases}$$

Így a $\sigma_\lambda(x) = \sigma(\lambda(y^T x + \theta) + \phi)$ függvény pontonként és korlátosan tart a

$$\gamma(x) = \begin{cases} 1 & \text{ha } y^T x + \theta > 0, \\ 0 & \text{ha } y^T x + \theta < 0, \\ \sigma(\phi) & \text{ha } y^T x + \theta = 0. \end{cases}$$

függvényhez, ha $\lambda \rightarrow +\infty$. Legyen $\Pi_{y,\theta}$ az $\{x|y^T x + \theta = 0\}$ által definiált hipertér és $H_{y,\theta}$ a $\{x|y^T x + \theta > 0\}$ által meghatározott nyílt féltér. Így Lebesgue korlátos konvergencia tétele miatt

$$\begin{aligned} 0 &= \int_{\mathcal{E}^n} \sigma_\lambda(x) d\mu(x) = \int_{\mathcal{E}^n} \gamma(x) d\mu(x) \\ &= \sigma(\phi) \mu(\Pi_{y,\theta}) + \mu(H_{y,\theta}). \end{aligned}$$

Most megmutatjuk, hogy ha a féltér mértéke 0, akkor a mértéknek kell 0-nak lennie. Ez triviális lenne, ha μ pozitív mérték lenne, de most nem az.

Rögzítsük y -t. Egy korlátos, mérhető h függvényre definiáljuk az F lineáris funkcionált az

$$F(h) = \int_{\mathcal{E}^n} h(y^T x) d\mu(x)$$

összefüggésnek megfelelően. Ekkor F egy korlátos funkcionál $L^\infty(\mathbb{R})$ -en, mivel μ egy véges előjeles mérték. Legyen h a $[\theta; \infty)$ intervallum indikátor függvénye, vagyis $h(u) = 1$, ha $u \geq \theta$ és $h(u) = 0$, ha $u < \theta$. Így

$$F(h) = \int_{\mathcal{E}^n} h(y^T x) d\mu(x) = \mu(\Pi_{y,-\theta}) + \mu(H_{y,-\theta}) = 0.$$

Hasonlóan, $F(h) = 0$, ha h a $(\theta; \infty)$ nyílt intervallum indikátor függvénye. A linearitás miatt $F(h) = 0$ bármelyik intervallum indikátor függvényére és így bármilyen egyszerű függvényre (az intervallumok indikátorfüggvényeinek összegére is). Mivel az egyszerű függvények sűrűek $L^\infty(\mathbb{R})$ -ben, így $F = 0$.

A korlátos mérhető $s(u) = \sin(m \cdot u)$ és $c(u) = \cos(m \cdot u)$ függvények mellett minden m -re:

$$F(s + ic) = \int_{\mathcal{E}^n} \cos(m^T x) + i \sin(m^T x) d\mu(x) = \int_{\mathcal{E}^n} \exp(im^T x) d\mu(x) = 0$$

teljesül. Ennélfogva μ Fourier transzformáltja 0, így μ -nek magának kell nullának lennie. Így a σ függvény diszkriminátor. \square

6.2. Neurális hálók alkalmazása

Ebben a részben az eddig eredményeket felhasználjuk a neurális hálók területén. A (6.1.2) tételből és a (6.1.3) lemmából következik, hogy az egy rejtett réteggel rendelkező neurális hálók egy tetszőleges szigmoid függvény segítségével tetszőleges pontossággal tudnak közelíteni folytonos függvényeket, feltéve hogy nem szabjuk meg a neuronok számát és a súlyok nagyságát.

6.2.1. Tétel. Legyen σ tetszőleges folytonos szigmoid függvény. Ekkor a

$$G(x) = \sum_{j=1}^N \alpha_j \sigma(y_j^T x + \theta_j)$$

alakú véges összegek sűrűek $\mathcal{C}(\mathcal{E}^n)$ -ben, vagyis tetszőleges adott $f \in \mathcal{C}(\mathcal{E}^n)$ függvényre és $\varepsilon > 0$ -ra létezik egy fenti alakú összeg, amelyre

$$|G(x) - f(x)| < \varepsilon$$

teljesül minden $x \in \mathcal{E}^n$ -re.

Bizonyítás. Az (6.1.2) tétel és az (6.1.3) lemma egyértelmű következménye, mivel a folytonos szigmoid függvények kielégítik a lemma feltételeit. \square

Most megmutatjuk a tétel azon következményeit, amelyek a döntési régiókhoz kapcsolódnak. Legyen m a Lebesgue mérték \mathcal{E}^n -n. Legyen P_1, P_2, \dots, P_k a \mathcal{E}^n egy partíciója k darab diszjunkt mérhető altérre. Definiáljuk az f döntési függvényt a következőképpen:

$$f(x) = j \quad \text{akkor és csak akkor, ha } x \in P_j.$$

Erre a függvényre úgy is tekinthetünk, mint osztályozó döntési függvényre, vagyis, ha $f(x) = j$, akkor tudjuk, hogy $x \in P_j$, így x -et eszerint osztályozhatjuk.

6.2.2. Tétel. Legyen σ egy folytonos szigmoid függvény, f pedig \mathcal{E}^n egy véges mérhető partíciójának döntési függvénye. Bármely $\varepsilon > 0$ -ra létezik egy

$$G(x) = \sum_{j=1}^N \alpha_j \sigma(y_j^T x + \theta_j)$$

alakú véges összeg és egy $D \subset \mathcal{E}^n$ úgy, hogy $m(D) \geq 1 - \varepsilon$ és

$$G(x) - f(x) < \varepsilon \quad x \in D\text{-re.}$$

Bizonyítás. A bizonyításhoz szükségünk lesz a következő tételre:

6.2.3. Lusin tétel. Tegyük fel, hogy f egy komplex mérhető függvény X -en, $\nu(A) < \infty$, $f(x) = 0$ ha $x \notin A$ és $\varepsilon > 0$. Ekkor létezik egy $g \in \mathcal{C}(X)$, hogy

$$\nu(x : f(x) \neq g(x)) < \varepsilon$$

teljesül. Továbbá megállapíthatjuk azt is, hogy

$$\sup_{x \in X} |g(x)| \leq \sup_{x \in X} |f(x)|.$$

Lusin tétele miatt létezik egy folytonos h függvény és egy D halmaz melyre $m(D) \geq 1 - \varepsilon$ így $f(x) = h(x)$ $x \in D$ -re. Most h folytonos, így a (6.2.1) tétel miatt létezik egy G véges összeg amely kielégíti a $|G(x) - h(x)| < \varepsilon$ egyenlőtlenséget minden $x \in \mathcal{E}^n$ -re. Így

$$|G(x) - f(x)| = |G(x) - h(x)| < \varepsilon$$

tejesül $x \in D$ -re. \square

A folytonosság miatt, mindig lesznek helytelenül osztályozott pontok, de ezen pontok mértéke tetszőlegesen kicsire viaszaszorítható. Ezt a döntési problémát egy zárt $D \subset \mathcal{E}^n$ halmazon vizsgálva kissé átalakíthatjuk az approximációs eljárást. Legyen az f a D halmaz indikátor függvénye. Mi egy olyan összeget szeretnénk találni, amely az (6.1)-es összefüggésben szerepel, hogy ezt a döntési függvényt közelíthessük. Legyen

$$\Delta(x, D) = \min\{|x - y|, y \in D\},$$

így $\Delta(x, D)$ az x folytonos függvénye. Legyen

$$f_\varepsilon(x) = \max \left\{ 0, \frac{\varepsilon - \Delta(x, D)}{\varepsilon} \right\},$$

így $f_\varepsilon(x) = 0$ azokra az x pontokra amelyek D -től nagyobb távolságra vannak, mint ε és $f_\varepsilon = 1$, ha $x \in D$, továbbá megjegyezzük azt is, hogy f_ε folytonos x -ben. A (6.2.1) tétel szerint találunk egy az (6.1)-es összefüggésnek megfelelő $G(x)$ -et, amivel $|G(x) - f_\varepsilon| < 1/2$. Ezt a $G(x)$ -et úgy használjuk, mint egy közelítő döntési függvényt: $G(x) < 1/2$ esetén azt gondoljuk, hogy $x \in D^C$, míg $G(x) \geq 1/2$ esetén azt, hogy $x \in D$. Ez a döntési eljárás jó az $x \in D$ pontokra és azokra is, amelyek legalább ε távolságra vannak D -től. Ha az x pont ε távolságon belül van D -től, akkor az osztályozás $G(x)$ választásától függ. Ez tehát azt mondja, hogy a döntési területen belüli vagy az attól elég távol levő pontok helyesen osztályozhatók. Nem úgy mint a (6.2.2) tétel, ami azt mondja, hogy létezik egy neurális háló, ami a helytelenül osztályozott pontok mértékét tetszőlegesen lecsökkentheti, de azok elhelyezkedését nem garantálja.

6.3. Más aktivációs függvények

Ebben a részben azokat az eddigiekhez hasonló eredményeket vizsgáljuk, amiket nem szigmoid aktivációs függvényekkel értek el. A nem folytonos szigmoid függvényeket nem használják olyan gyakran, mint a folytonosakat, mert ezekhez még nem találtak megfelelő tanítási algoritmusokat. Mivel azonban szoros kapcsolatban állnak a perceptronnal és a Gamma hálókkal ezért mégis nagyon fontosak.

Tegyük fel, hogy σ egy korlátos és mérhető szigmoid függvény, majd tekintsük a következő tételt, amely a korábban látott (6.2.1) tétel egy analógja:

6.3.1. Tétel. *Legyen σ egy korlátos, mérhető szigmoid függvény. Ekkor a*

$$G(x) = \sum_{j=1}^N \alpha_j \sigma(y_j^T x + \theta_j)$$

alakú összeg sűrű $L^1(\mathcal{E}^n)$ -ben, vagyis adott egy tetszőleges $f \in L^1(\mathcal{E}^n)$ függvény és egy $\varepsilon > 0$ szám, ekkor létezik egy olyan, az előzőeknek megfelelő $G(x)$ összeg, amelyre

$$\|G - f\|_{L^1} = \int_{\mathcal{E}^n} |G(x) - f(x)| dx < \varepsilon.$$

A tétel bizonyítása a (6.1.2) és a (6.2.1) tételek bizonyításából következik néhány változtatással. A folytonos függvényeket integrálható függvényekre cseréljük és az $L^\infty(\mathcal{E}^n)$ helyett a duálisát, $L^1(\mathcal{E}^n)$ -et használjuk. A diszkriminátorságnak a következő feltétel felel meg: ha $h \in L^\infty(\mathcal{E}^n)$ -ra

$$\int_{\mathcal{E}^n} \sigma(y^T x + \theta) h(x) dx = 0$$

teljesül minden y és θ -ra, akkor következik, hogy $h(x) = 0$ majdnem mindenütt. Az általános szigmoid függvények diszkriminátorok ebben az értelemben, ahogy láttuk a (6.1.3) lemmában, mert $h(x)dx$ mértéke $M(\mathcal{E}^n)$ -hez tartozik.

Az L^1 -beli konvergenciából következik a mértékben való konvergencia, így van egy analógja a (6.2.2) tételnek is:

6.3.2. Tétel. *Legyen σ egy általános szigmoid függvény és f a döntési függvénye \mathcal{E}^n egy tetszőleges véges, mérhető partíciójának. Tetszőleges ε -ra létezik egy*

$$G(x) = \sum_{j=1}^N \alpha_j \sigma(y_j^T x + \theta_j)$$

alakú véges összeg és egy $D \subset \mathcal{E}^n$ halmaz, amire

$$m(D) \geq 1 - \varepsilon \quad \text{és}$$

$$|G(x) - f(x)| < \varepsilon$$

teljesül $x \in D$ esetén.

A Stone-Weierstrass tétel egyszerű alkalmazásával megmutatható, hogy nemcsak a (6.1.2) tételben szereplő aktivációs függvények bírnak az ott leírt approximációs tulajdonságokkal, hanem más lehetséges aktivációs függvények is. A tételbeliek tartalmazzák a \sin és \cos függvényeket. Ez kézenfekvő, hiszen a $\sin(kt)$ és a $\cos(kt)$ függvények lineáris kombinációi generálják az összes véges trigonometrikus polinomot, amelyekről tudjuk, hogy teljesek $C(\mathcal{E}^n)$ -ben. A trigonometrikus polinomok ezt a fajta teljességet implicit módon felhasználtuk a (6.1.3) lemmában, amikor a Fourier transzformált kölcsönösen egyértelmű leképezési tulajdonságát alkalmaztuk.

Egy másik klasszikus példa az exponenciális függvények, vagyis az $\exp(at)$ alakú függvények használata. A bizonyítás ebben az esetben is a Stone-Weierstrass felhasználásával következik.

Egy teljesen más típusa az aktivációs függvényeknek, melyeket Wiener Tauber féle tételéhez köthetünk, szintén rendelkezik a teljesség tulajdonságával $L^1(\mathcal{E}^n)$ -ben.

6.3.3. Wiener Tauber féle tétele. Legyen f egy $L^1(\mathbb{R})$ -beli integrálható függvény. Az $fa(x) = f(x+a)$ eltolások összes véges lineáris kombinációi által generált altér akkor és csak akkor sűrű $L^1(\mathbb{R})$ -ben, ha f Fourier transzformáltja nem nulla.

Tegyük fel például, hogy σ egy tetszőleges $L^1(\mathbb{R})$ -beli függvény, melynek integrálja nem 0. Ekkor a (6.1)-es formulának megfelelő összegek sűrűek $L^1(\mathbb{R}^n)$ -ben, ahogy azt a következőkben láthatjuk.

A (6.1.2) tétel teljes mértékben átvezethető ide, ha a $C(\mathcal{E}^n)$ -et lecseréljük $L^1(\mathcal{E}^n)$ -re, $M(\mathcal{E}^n)$ -t pedig a megfelelő duális térre, $L^\infty(\mathcal{E}^n)$ -re. A (6.2.2) tétel is érvényben marad, ha megtudjuk mutatni, hogy egy integrálható σ függvény, melynek integrálja nem 0 diszkriminátor, vagyis ha

$$\int_{\mathcal{E}^n} \sigma(y^T x + \theta) h(x) dx = 0 \quad (6.3)$$

minden y -ra és θ -ra teljesül, akkor következik, hogy $h = 0$. Ehhez a következők szerint járunk el. Mint a (6.1.3) lemmában, itt is definiáljuk a korlátos, lineáris funkcionált, F -et $L^1(\mathbb{R})$ -ben az

$$F(g) = \int_{\mathcal{E}^n} g(y^T x) h(x) dx$$

összefüggés szerint. (Itt megjegyezzük, hogy az integrál létezik, ha véges \mathcal{E}^n -en és ha h korlátos \mathcal{E}^n -en. Speciálisan, ha $g \in L^1(\mathbb{R})$, akkor $g(y^T x) \in L(\mathcal{E}^n)$ tetszőleges y -ra.) Láthatjuk a $g_{\theta,s}(t) = \sigma(st + \theta)$ jelölés mellett, hogy

$$F(g_{\theta,s}) = \int_{\mathcal{E}^n} \sigma((sy)^T x + \theta) h(x) dx = 0$$

teljesül, így F megszűntet minden eltolást és skálázást $g_{0,1}$ -en. Legyen \tilde{f} az f függvény Fourier transzformáltja. A standard Fourier transzformált argumentumai alapján $\tilde{g}_{\theta,s}(z) = \exp(iz, \theta/s) \tilde{g}(z/s)/s$. Az s szerinti skálázás miatt, az egyetlen z érték amelyre a minden $g_{\theta,s}$ Fourier transzformáltja eltűnhet, az a $z = 0$. De feltesszük, hogy $\int_{\mathbb{R}} \sigma(t) dt = \tilde{g}_{0,1}(0) \neq 0$. Wiener Tauber féle tétele alapján a $g_{\theta,s}$ függvények által generált altér sűrű $L^1(\mathbb{R})$ -ben. Az $F(g_{\theta,s}) = 0$ összefüggés miatt $F = 0$ mindenképpen kell, ez implikálja azt, hogy

$$F(\exp(imt)) = \int_{\mathcal{E}^n} \exp(imt) h(t) dt = 0$$

minden m -re és emiatt lesz h Fourier transzformáltja 0, ennél fogva h maga is 0 lesz.

Wiener Tauber féle tétele további különleges aktivációs függvényekhez vezet, amelyek rendelkeznek az $L^1(\mathcal{E}^n)$ -beli teljesség tulajdonságával.

Tekintsük például a következő n -változós aktivációs függvényt. Vegyünk egy rögzített téglalapot, melynek n oldala egybeesik \mathbb{R}^n koordinátatengelyeivel. Legyen $\sigma(x) = 1$, ha x a téglalapon belül van és 0 minden egyéb esetben. Legyen U egy $n \times n$ -es ortogonális mátrix és $y \in \mathbb{R}^n$. Most $\sigma(Ux + y)$ egy tetszőleges irányú téglalap indikátor függvénye.

Így a

$$\sum_{j=1}^N \alpha_j \sigma(U_j x + y_j)$$

alakú összeg sűrű lesz $L^1(\mathbb{R}^n)$ -ben. Ez Wiener Tauber féle tételének direkt alkalmazásából és abból következik, hogy a σ Fourier transzformáltja eltűnik egy olyan \mathbb{R}^n -beli hálón, mely nem tartalmazza az origót. Ezen hálók összes lehetséges elforgatásából alkotott metszet üres és így σ az eltolt és elforgatott változataival együtt generál egy $L^1(\mathbb{R}^n)$ -ben sűrű teret.

7. fejezet

Az R program neurális hálókkal foglalkozó csomagjainak bemutatása

Ebben a fejezetben bemutatunk néhány beépített csomagot az R-ben, amelyek neurális hálókkal foglalkoznak.

7.1. Az nnet

Az nnet-et loglineáris modellekhez és olyan feed-forward neurális hálókhoz használhatjuk, melyeknek egy rejtett rétege van. Létrejött Brian Ripley nevéhez köthető. A csomagban lévő függvények rugalmasságának köszönhetően, mi magunk alakíthatjuk a legjobb vagy legoptimálisabb modelleket a változó paraméterektől a tanítás folyamatáig.

Ahhoz, hogy bemutassuk a csomag működését, lássunk egy példát az `iris` adatbázissal, amely a nőszirm csésze és szirmleveleiről tartalmaz adatokat:

```
> ir <- rbind(iris3[, ,1], iris3[, ,2], iris3[, ,3])
> targets <- class.ind( c(rep("s", 50), rep("c", 50), rep("v", 50)) )
> samp <- c(sample(1:50,25), sample(51:100,25), sample(101:150,25))
> ir1 <- nnet(ir[samp,], targets[samp,], size = 2,
              rang = 0.1, decay = 5e-4, maxit = 200)
```

Itt az első három sorban megadjuk, hogy az adatbázis mely részeit szeretnénk használni. Ahogy láthatjuk csak a negyedik sorban használjuk az `nnet` függvényt, amely a következő argumentumokkal rendelkezik:

- először az input adatokat adjuk meg egy mátrixban vagy dataframe-ben
- a második helyen a célértékeket adhatjuk meg ugyanúgy, mint az inputot,
- a `size=` argumentummal a rejtett rétegen lévő neuronok számát határozhatjuk meg,
- a `rang=0.1` paranccsal megadjuk, hogy a függvény a kezdő súlyokat a $[-0, 1; 0, 1]$ intervallumból választja véletlen módon,

- a `decay` argumentum a hiba lehetséges maximumát határozza meg,
- végül a `maxit` változóval az iterációk számát adhatjuk meg.

A függvény meghívása után eredményül a következőket kapjuk:

```
# weights:  19
initial  value 56.789137
iter   10 value 48.733287
iter   20 value 20.996305
iter   30 value 18.707109
iter   40 value 18.613420
iter   50 value 18.241921
iter   60 value 18.218606
iter   70 value 18.210357
iter   80 value 18.207759
final   value 18.207633
converged
```

Tehát láthatjuk, hogy 19 darab súlyunk van, látjuk a kezdőértékünket és azt is, hogy az egyes iterációk után hogyan alakul ez az érték. Végül a végső értéket is megkapjuk és azt, hogy az iteráció konvergál. Ezeken kívül természetesen egyéb argumentumokat is adhatunk a függvénynek, így meghatározhatjuk, hogy milyen hibafüggvényt használjunk és a kezdő súlyokat is megadhatjuk. Az `ir1` paranccsal kiírathatjuk a neurális háló és a tanítása néhány tulajdonságát:

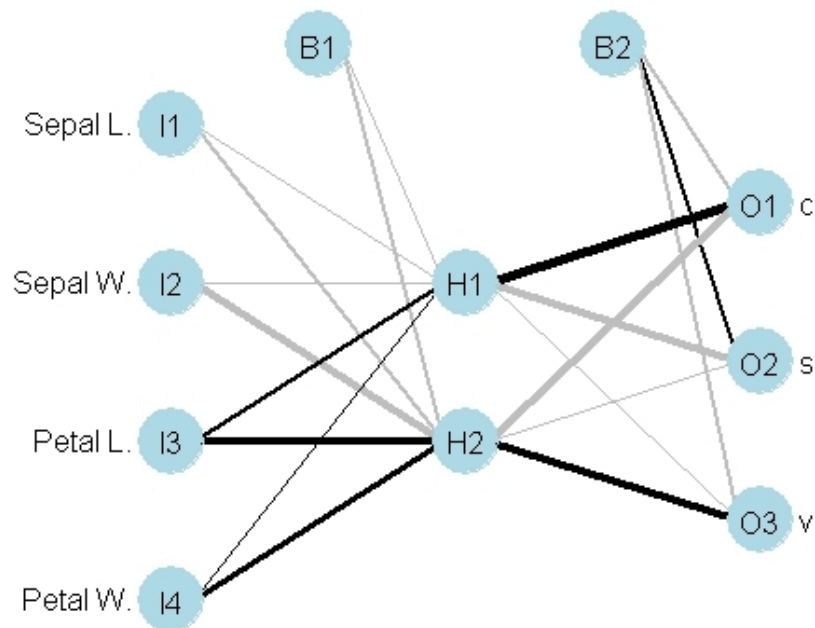
```
a 4-2-3 network with 19 weights
options were - decay=5e-04
```

Ahol 4-2-3 az egyes rétegeken lévő neuronok számát jelöli.

A csomag legfőbb hátránya, hogy nem tudjuk ábrázolni vele a modelleket. Létezik azonban egy olyan R-beli függvény, melynek segítségével kirajzoltathatjuk az `nnet` által kapott neurális hálót egy neurális értelmezési diagramként.

S. L. Özesmi és U. Özesmi az 1999-ben megjelent írásukban (*An artificial neural network approach to spatial habitat modeling with interspecific interaction. Ecological Modelling. 116:15-31.*) említik a neurális értelmezési diagrammot (NID), amellyel vizualizálni tudunk egy már betanított neurális hálót. Ez a diagram megmutatja a rétegek közötti kapcsolatokat is.

Ezt láthatjuk a következő 7.1 ábrán, melyen az input adatokat $I1, I2, I3, I4$, a kapott eredményeket pedig $O1, O2$ és $O3$ jelöli. A H jelű neuronok a rejtett réteget alkotják, $B1$ és $B2$ pedig a torzításnak felelnek meg. Amint láthatjuk, az éleket is két különböző színnel jelölik, fekete élek a pozitív súlyúak, a szürkék pedig a negatívak. Különböző a vastagságuk is, ez a súlyok egymáshoz viszonyított arányát jelöli.



7.1. ábra. Neurális értelmezési diagram

7.2. A neuralnet

Ez a csomag nagyon rugalmas, alkalmas a többrétegű perceptron tanítására is. Erre a célra több beépített algoritmus áll a rendelkezésünkre, köztük a már sokszor említett backpropagation algoritmus is. A felhasználó képes különböző extra rejtett rétegeket beiktatni a neurális hálóba vagy csökkenteni a rétegenkénti neuron számot, ezzel szorítva vissza a számítási költségeket. Sikeresen alkalmazható például a biológiai gén-gén kapcsolatok részletes leírására, ezzel jól modellezve komplexebb betegségeket is.

Bemutatunk egy példát, ehhez az **infern** nevű adatbázist használjuk, amiben 248 megfigyelés szerepel: 83 eset, ők terméketlenek és 165, akik nem terméketlenek. Ezek mellett találunk benne még egyéb változókat is, mint a vizsgált nők kora, a születések, az abortuszok és a spontán vetélések száma. A spontán vetélések és az abortuszok száma a régebbi esetekre (a vizsgálat előtti) vonatkozik. Mindkét változó a 0, 1 és 2 értékeket veheti fel, attól függően, hogy 0, 1 vagy 2 esetleg annál több vetélése volt régebben. A program segítségével megpróbáljuk modellezni a meddőség kapcsolatát a másik négy változóval (kor, születések száma, abortusz és spontán vetélés). Ebben az esetben az utóbbi négy változó értékei lesznek az input adatok, a meddőség ténye pedig az output. Mivel az output értékek binárisak lesznek, ezért a logisztikus függvényt választhatjuk aktivációs függvénynek, hiba-függvénynek pedig a kereszt-entrópiát. Jelen esetben egy olyan neurális hálóval dolgozunk, melynek rejtett rétegén két neuron van. A neuralnet által való tanítás a következő módon zajlik:

```
> library(neuralnet)
```

```
> nn <- neuralnet(case~age+parity+induced+spontaneous,
                  data=infert, hidden=2, err.fct="ce",
                  linear.output=FALSE)
```

- Először megadjuk az input és output adatokat a programnak melyeket a ~ karakter választ el egymástól (először az output aztán az input adatok).
- A data= paranccsal adhatjuk meg, hogy milyen adatbázison dolgozzunk.
- A hidden=-nel pedig azt, hogy a rejtett rétegen hány neuron legyen.
- Az err.fct=" " paranccsal a hibafüggvény típusát határozhatjuk meg.
- A linear.output két értéket vehet fel: TRUE vagy FALSE, attól függően, hogy az aktivációs függvény alkalmazható-e az output réteg neuronjain vagy sem.

Az alapvető információkat a tanítás folyamatáról és a már betanított hálóról a program elmentette az nn nevű fájlba, ezeket a következő módon kaphatjuk meg:

```
>nn
Call: neuralnet(formula = case ~ age + parity + induced + spontaneous,
                 data = infert, hidden = 2,
                 err.fct = "ce",
                 linear.output = FALSE)

1 repetition was calculated.

Error Reached Threshold Steps
1 125.2046588    0.009074149749 1862
```

A következő paranccsal kiírathatjuk összegezve a főbb eredményeket:

```
> nn$result.matrix

error                125.204658773160
reached.threshold    0.009074149749
steps                1862.000000000000
Intercept.to.1layhid1 1.789036305611
age.to.1layhid1      1.656148885899
parity.to.1layhid1   1.042102710960
induced.to.1layhid1  1.405403753280
spontaneous.to.1layhid1 0.071859466117
Intercept.to.1layhid2 -5.487322577575
```

age.to.1layhid2	0.110802632432
parity.to.1layhid2	-1.624494364507
induced.to.1layhid2	2.047877005824
spontaneous.to.1layhid2	3.109419216813
Intercept.to.case	-0.460008692727
1layhid.1.to.case	-1.349923390663
1layhid.2.to.case	5.770172858272

Ezek között megtalálhatjuk, hány lépésből állt a tanítás, mekkora a hiba, sőt az input, a rejtett és az output réteg neuronjai közötti összes élen lévő súlyt is láthatjuk.

A neurális háló outputját, vagyis a kiszámított értékeket is kiírathatjuk a következő parancsokkal:

```
> out <- cbind(nn$covariate, nn$net.result[[1]])
> dimnames(out) <- list(NULL, c("age", "parity", "induced",
                                spontaneous", "nn-output"))

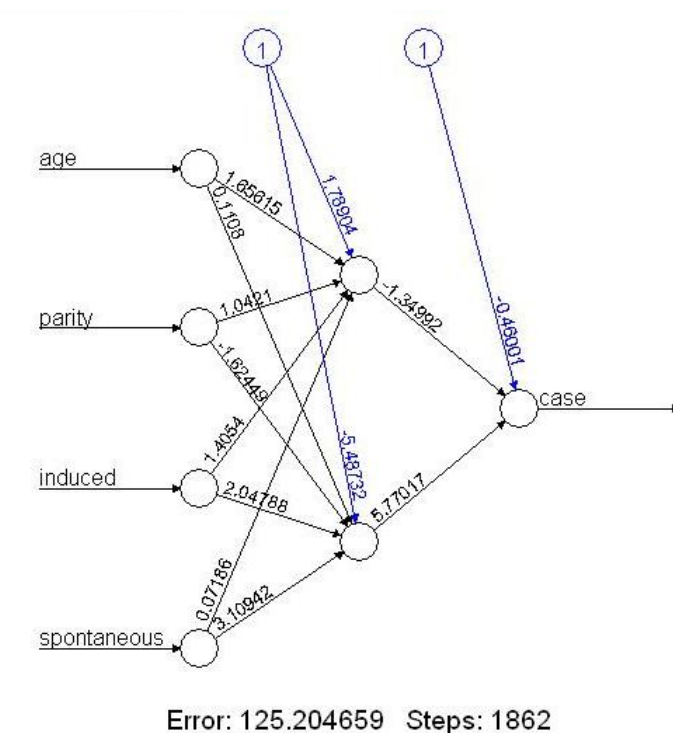
> head(out)
```

az ezután kapott eredmények így néznek ki:

	age	parity	induced	spontaneous	nn-output
[1,]	26	6	1	2	0.1525647111
[2,]	42	1	1	0	0.6204697294
[3,]	39	6	2	0	0.1414107193
[4,]	34	4	2	0	0.1521416837
[5,]	35	3	1	1	0.3546488147
[6,]	36	4	2	1	0.4974522710

7.2.1. Ábrázolás

A kapott eredmények vizualizálására több lehetőségünk is van. Ha a betanított neurális hálót szeretnénk kirajzoltatni, akkor a `plot(nn)` parancsot kell használnunk. A kapott eredményt a 7.2 kép mutatja. Itt láthatjuk, hogy az egyes neuronok hogyan kapcsolódnak egymáshoz, sőt az éleken a súlyok is fel vannak tüntetve. A program által kapott ábrán az is szerepel, hogy mekkora a hiba és hogy a tanítás hány lépésből állt.

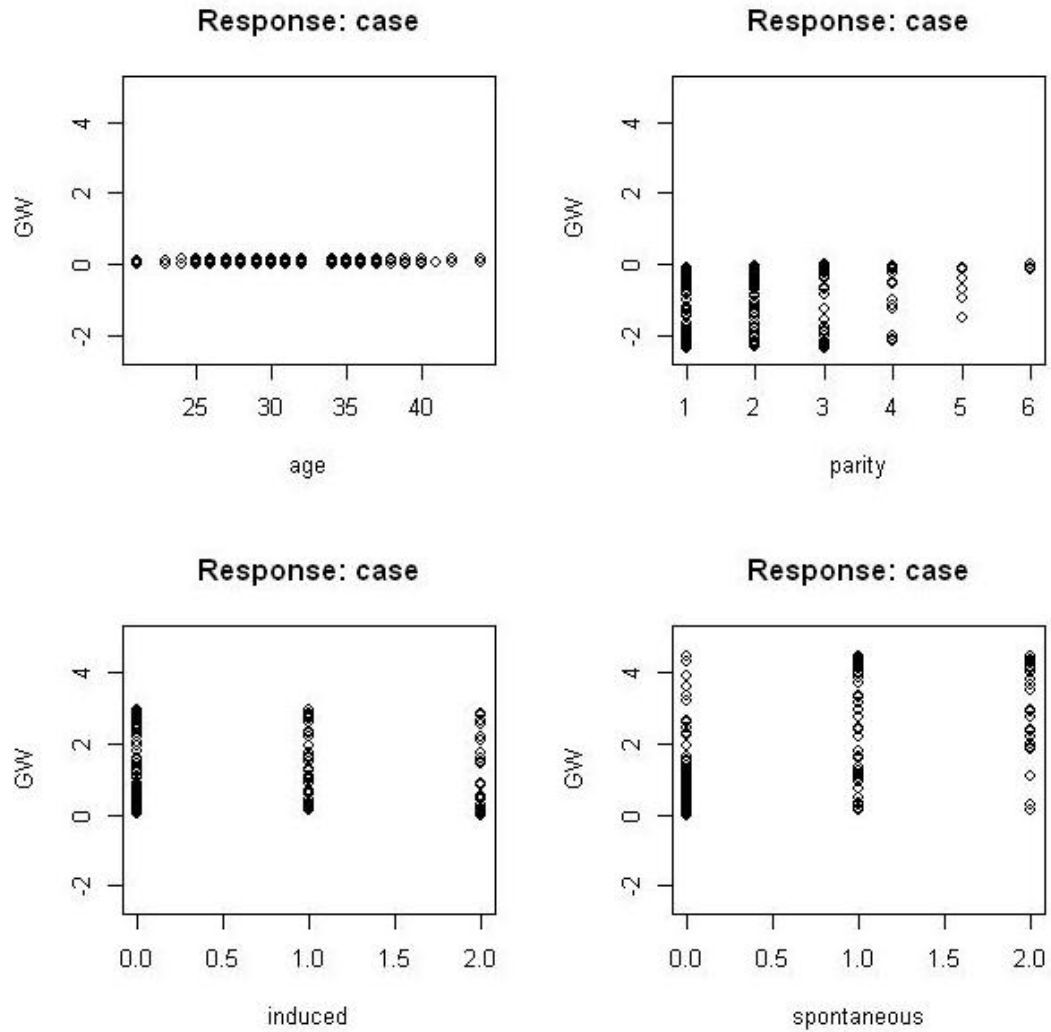


7.2. ábra. A neurális háló

A másik lehetséges mód, hogy a generált súlyokat ábrázoljuk az egyes input adatok és az output között, ezt a következő módon tehetjük meg:

```
> par(mfrow=c(2,2))
> gwplot(nn,selected.covariate="age", min=-2.5, max=5)
> gwplot(nn,selected.covariate="parity",min=-2.5, max=5)
> gwplot(nn,selected.covariate="induced",min=-2.5, max=5)
> gwplot(nn,selected.covariate="spontaneous", min=-2.5, max=5)
```

A kapott grafikonokat mutatja a 7.3 ábra. Az elsőn a megfigyelt nők kora, a másodikon a szülések a száma, a harmadikon az abortuszok, a negyediken pedig a spontán vetélések száma és a meddőség ténye, vagyis az output közötti generált súlyokat mutatja:



7.3. ábra. A generált súlyok az egyes input adatok függvényében

7.3. Az RSNNs

Ezt a csomagot a stuttgarti egyetemen fejlesztették, innen ered a neve is: Stuttgart Neural Network Simulator. Ez az átfogó alkalmazás a neurális hálók építésére, tanítására és tesztelésére is megoldást ad, a ma ismert programok közül az egyik legteljesebb, legmegbízhatóbb és leggyorsabb megoldást adja a témával kapcsolatos problémákra.

Az RSNNs magasszintű felhasználói felülete kényelmes utat biztosít ahhoz, hogy megismerjük a neurális hálók leggyakoribb fajtáit. Sok fajta modellt támogat (mint például a többrétegű perceptron, az ART1, ART2, ARTMAP, a SOM, az RBF és a Hopfield hálózatokat is.) A neurális hálók típusai számos dologban eltérnek egymástól, például a működésük milyenségében, a szerkezetükben vagy akár a tanításuk típusában is. Emiatt többféle tanítási algoritmus és aktivációs függvény is a felhasználó rendelkezésére áll.

A felhasználó felület minden modellenél nagyon hasonló:


```
> mlp(x, y, size = c(5), maxit = 100,
      initFunc = "Randomize_Weights",
      initFuncParams = c(-0.3, 0.3),
      learnFunc = "Std_Backpropagation",
      learnFuncParams = c(0.2, 0),
      updateFunc = "Topological_Order",
      updateFuncParams = c(0),
      hiddenActFunc = "Act_Logistic", shufflePatterns = TRUE,
      linOut = FALSE, inputsTest = NULL, targetsTest = NULL)
```

- Tanító adatok: x, y . Itt x egy mátrix vagy egy vektor, amely az input adatokat tartalmazza. Ha felügyelt tanítási algoritmust választunk, akkor y -al jelölt célértékeket is meg kell adnunk.
- A hálózat paraméterei: az *mlp*-nek (többrétegű perceptron) csak egy paramétert kell megadnunk ez pedig a mérete (**size**), itt adhatjuk meg a rejtett rétegen lévő neuronok számát.
- Az iterációk száma (**maxit**): itt az iterációk vagyis az epochok számát adhatjuk meg.
- Inicializáló függvény: az **initFunc** és **initFuncParams** paraméterekkel adhatjuk meg. Itt tulajdonképpen a neurális háló súlyait határozhatjuk meg. (Fent véletlenszerűen adtuk meg a súlyokat.)
- Tanítási algoritmus: **learnFunc** és **learnFuncParams**, a neurális hálónk egyik legmeghatározóbb tulajdonsága. A különböző algoritmusokhoz különböző paramétereket kell megadnunk. Fent a standard backpropagation algoritmust használtuk, ehhez két paramétert adhatunk meg: a tanulási paramétert és a d_{max} értéket, mely a célértékek és az output maximális különbsége.
- Az **updateFunc** és az **updateFuncParams** segítségével meghatározhatjuk, hogy milyen szabály alapján változtassuk az algoritmus alatt a súlyokat.
- A módszer speciális paraméterei: **hiddenActFunc**, **shufflePatterns**, és **lineOut**. Itt adhatjuk meg, hogy a rejtett rétegeken (**hiddenActFunc**) és az output rétegen (**lineout**) milyen aktivációs függvényeket használjunk és hogy a mintánkat véletlenítse-e a program.
- Tesztadatok (**inputsTest**, **targetsTest**): itt adhatjuk meg, hogy mely adatokon teszteljük le a hálónkat.

Ha nem többrétegű perceptront akarunk használni, akkor értelemszerűen nem az **mlp()** függvényt hívjuk, hanem például az **art1()**, az **artmap()**, az **elman()** vagy a **jordan()**

függvényeket. A R-ben beépített adatbázisokat is találunk, melyeken könnyen alkalmazhatjuk a függvényeket. Lássunk egy olyan példát, amelyben a már korábban látott `iris` adatbázisra alkalmazzuk a programot:

```
data("iris")

> iris <- iris[sample(1:nrow(iris),length(1:nrow(iris))),1:ncol(iris)]
> irisValues <- iris[, 1:4]
> irisTargets <- iris[, 5]
> irisDecTargets <- decodeClassLabels(irisTargets)
> iris <- splitForTrainingAndTest(irisValues,irisDecTargets,ratio = 0.15)
> iris <- normTrainingAndTestSet(iris)
> model <- mlp(iris$inputsTrain,iris$targetsTrain,size = 5,
               learnFuncParams = 0.1,maxit = 60,
               inputsTest = iris$inputsTest,
               targetsTest = iris$targetsTest)
> prediction <- predict(model,iris$inputsTest)
```

A `model` fájlba elmentettük a neurális háló és a tanításának legfontosabb tulajdonságait, melyet a függvény meghívásakor határoztunk meg, ezeket a következő módon hívhatjuk elő:

```
> model

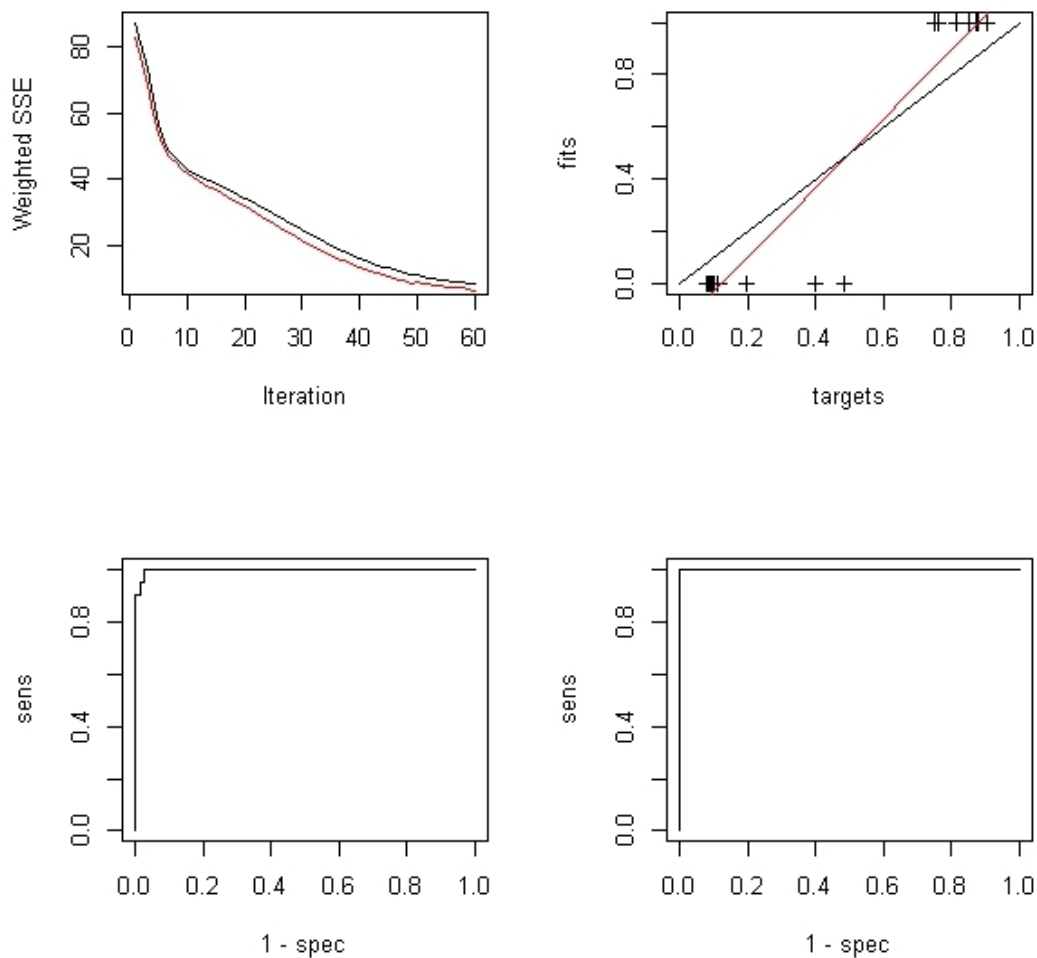
Class: mlp->rsnns
Number of inputs: 4
Number of outputs: 3
Maximal iterations: 60
Initialization function: Randomize_Weights
Initialization function parameters: -0.3 0.3
Learning function: Std_Backpropagation
Learning function parameters: 0.1
Update function:Topological_Order
Update function parameters: 0
Patterns are shuffled internally: TRUE
Compute error in every iteration: TRUE
Architecture Parameters:
size
[1] 5
```

A neurális háló további meghatározó tulajdonságait is megkaphatjuk a `summary(model)` paranccsal, mint az egyes neuronokhoz tartozó aktivációs függvények típusa, a torzítások, valamint a neuronok között lévő összes kapcsolat súlya.

7.3.1. Ábrázolás

A kapott eredményeket a következő módon jeleníthetjük meg:

```
> plotIterativeError(model)
> plotRegressionError(prediction[, 2],iris$targetsTest[, 2],pch = 3)
> plotROC(fitted.values(model)[, 2],iris$targetsTrain[, 2])
> plotROC(prediction[, 2],iris$targetsTest[, 2])
```



7.4. ábra. A többrétegű perceptron eredményei

A 7.4 ábrán az `iris` adatbázissal tanított többrétegű perceptron eredményei láthatók. Az elsőn a iteratív hibák: feketével jelöltük a tanító és pirossal a teszt adatokon kapott hibát. A másodikon a teszt adatok regressziója van. A harmadikon és a negyediken egy-egy ROC görbe látható: ezek megmutatják, hogy mennyire sikerült helyesen osztályozni a pontokat. A harmadikon a tanító adatokra, a negyediken pedig a teszt adatokra vonatkoztatva. (Az ábrák alapján megállapíthatjuk, hogy itt viszonylag jól sikerült az osztályozás: a negyedik ábrán egy tökéletes, míg a harmadikon egy majdnem tökéletes osztályozás eredményét láthatjuk.)

8. fejezet

A neuralnet használata életminőségi adatokra

Ebben a fejezetben olyan életminőségi indikátorokat szeretnénk összefüggésbe hozni a kibocsátott nitrogén- és kénoxidok mennyiségével, mint a csecsemő halandóság, a születéskor várható élettartam, a GDP és a GDP növekedési ráta. Köztudott, hogy a levegő minősége nagy mértékben befolyásolja az adott területen élő populáció egészségét és így az életminőségére és a gazdaságára is hatással van.

A kén- és nitrogénoxidok az üzemanyag fogyasztás melléktermékei. A nitrogénoxidok a gépjárművek motorjai bocsájtják ki, a kibocsátott kénoxidok pedig a magas kéntartalmú szénkeletéssel keletkeznek, ez ipari folyamatokhoz köthető.

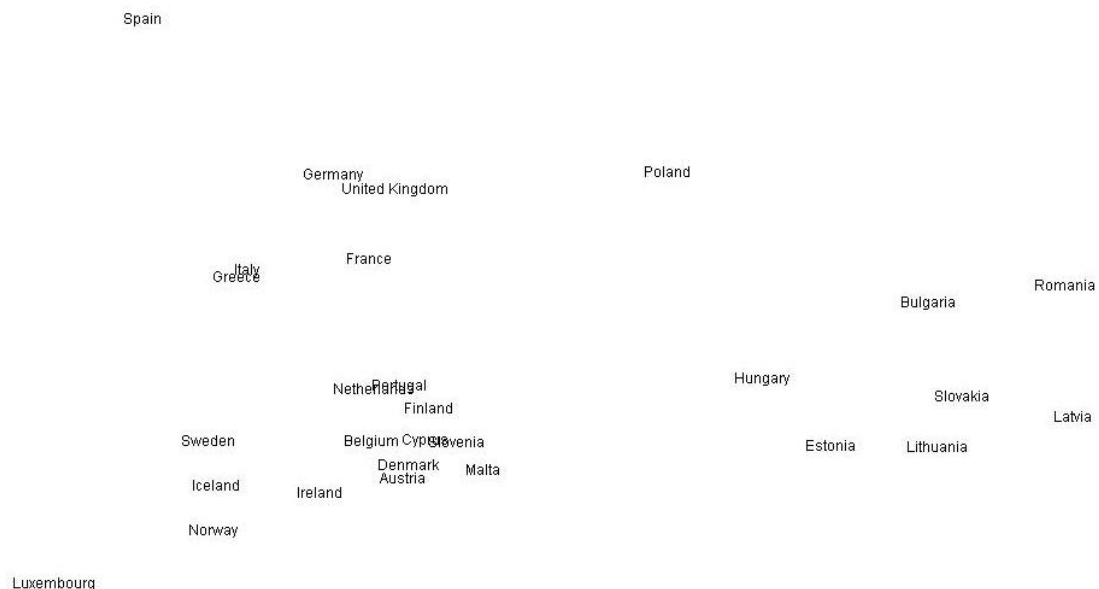
Az adatok európai országokra vonatkoznak és az eurastat.com 2012-es adatbázisaiból származnak. Az ötletet egy 2010-ben megjelent cikk adta, amely ugyanilyen, csak 2005-ből származó adatokat dolgoz fel (*Kyriaki Kitikidou, Lazaros Iliadis: Developing Neural Networks to Investigate Relationships Between Air Quality and Quality of Life Indicators*). A cikkel ellentétben mi azonban most osztályozni fogjuk az országokat a különböző adatok alapján. Ehhez az előbb bemutatott programcsomagokat fogjuk használni. Az adatbázisunkban tehát a következő életminőségi indikátorokat szerepelnek:

- születéskor várható élettartam: az élve születések halálozási életkorának átlaga;
- egészséges évek száma: az adott életkorú személy (esetünkben egy újszülött) várhatóan hány évet fog betegségtől mentesen eltölteni;
- csecsemő halandóság: 1000 élve születésre jutó 1 éves kor alatti halálozások éves száma;
- egy főre jutó GDP: (bruttó hazai termék) egy adott országban, adott idő alatt (általában egy év) előállított végső felhasználásra szánt javak, termékek és szolgáltatások összességének értéke egy főre leosztva;

- GDP növekedési ráta: azt mutatja meg, hogy egy adott országban az előző évhez képest mennyivel nőtt a bruttó hazai termék.

A táblázatban emellett a kibocsátott nitrogén- és kénoxidok mennyiségét tartalmazza (millió tonnában).

Az adatbázisunkat beolvasva az R-be elvégeztünk egy faktoranalízist, ez alapján az országokat két elkülönülő csoportba soroltuk, ezt láthatjuk a 8.1 ábrán. Az adatbázist



8.1. ábra. A táblázatban szereplő országok csoportosítva

ezután kiegészítjük a **regio** tulajdonsággal, annak alapján, hogy az adott országot melyik osztályba soroltuk. Ezután betöltjük a **neuralnet** csomagot, majd lefuttatjuk:

```
M<-neuralnet(regio~gdp.growth+gdp+hly+infmor+life.exp+sulphur+nitro,
              D,hidden=h)
```

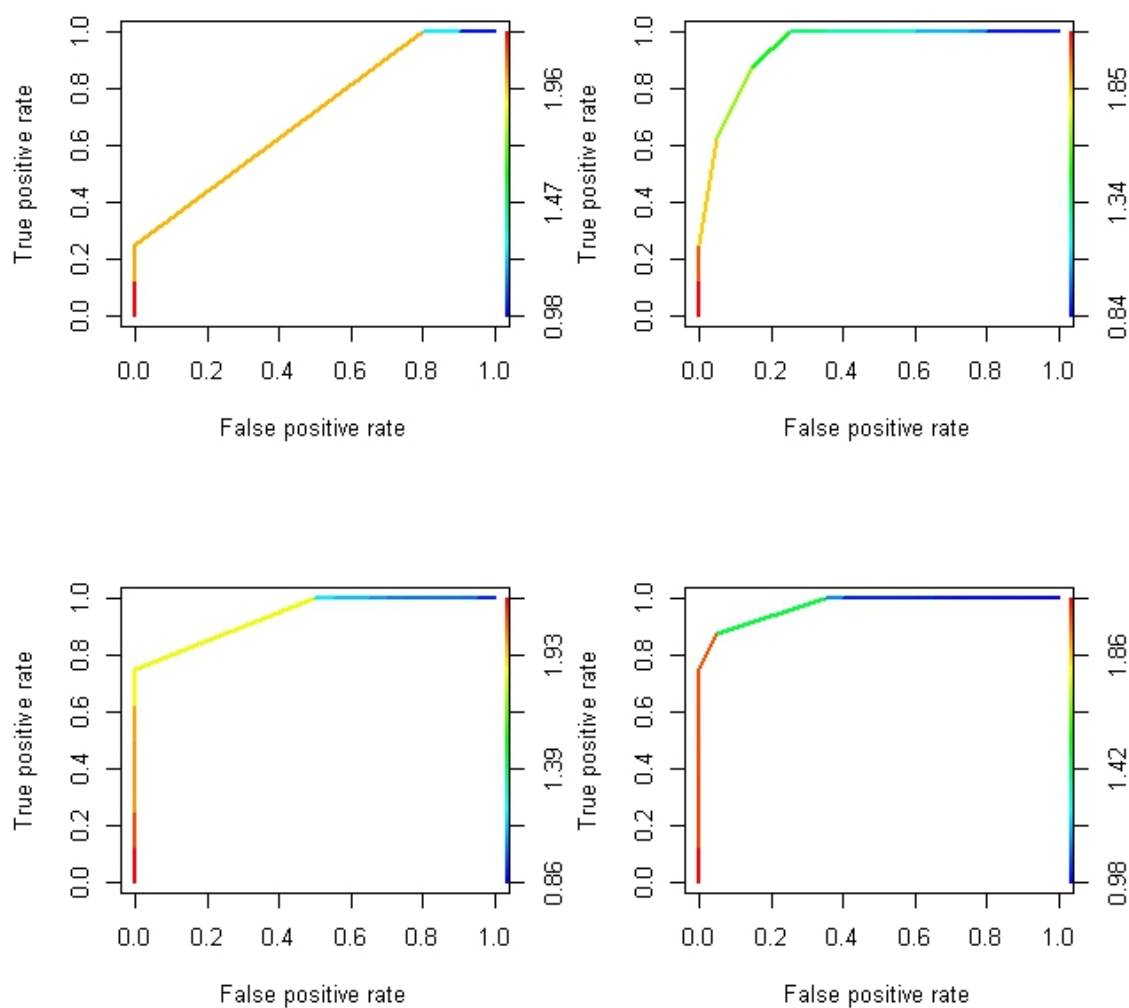
D-vel jelöljük az egész adatbázist, melynek a **regio** oszlopát állítottuk be célértékként, a többit pedig inputként kezeljük. A rejtett rétegen lévő neuronok számát **h**-val jelöljük, így a későbbiekben könnyen tudjuk majd változtatni. A tanítás és a neurális háló főbb tulajdonságai a következők:

```
Call: neuralnet(formula = regio ~ gdp.growth + gdp + hly +
                  infmor + life.exp + sulphur + nitro,
                  data = D, hidden = h)
```

```
1 repetition was calculated.
```

```

Error Reached Threshold Steps
1 2.434916844    0.009231110052    208
```

8.3. ábra. ROC görbék

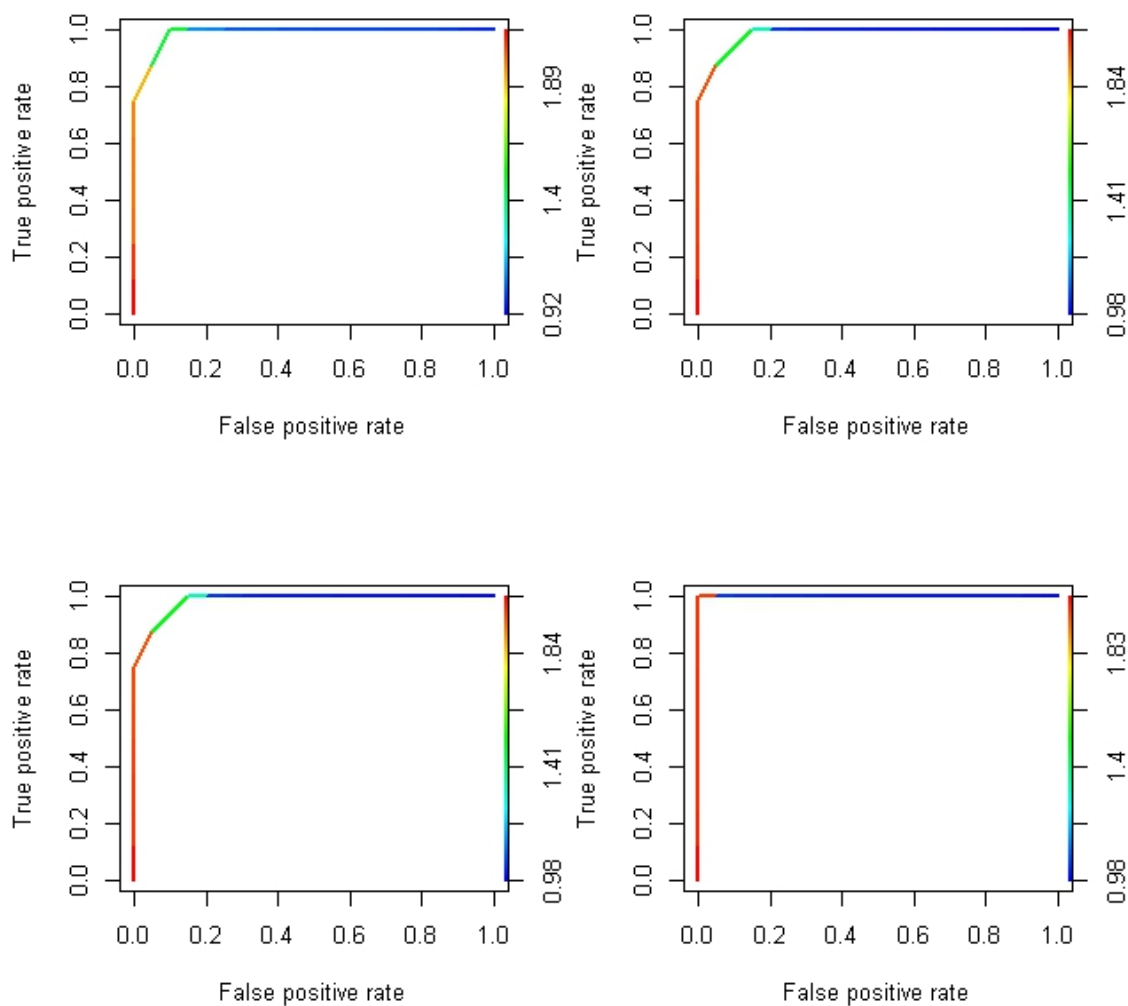
A ROC görbék alapján, ha a vágási szintet például 1,5-nek állítjuk, kiíráthatjuk, hogy pontosan hány országot osztályoztunk helyesen és hányat nem:

```
pred.num<-as.numeric(P$net.res<1.5)
table(D$regio,pred.num)
```

A rejtett rétegen 24 neuronnal bíró hálózattal ez így néz ki:

```
pred.num
  0  1
1  0 20
2  3  5
```

Tehát a 28 országból 23-at sikerült helyesen osztályozni és 5-öt nem, ami egy viszonylag jó eredmény.



8.4. ábra. ROC görbék

A 8.4 ábrán a rejtett rétegen 72, 96, 112 és 136 neuronnal kapott osztályozás eredményei láthatók. A neuronok számának növelésével az R-ben történt számítások időtartama is nőtt.

8.1. Az R-ben futtatott függvények és az általuk kapott eredmények

Ebben a részben láthatók részletesen azok a függvények és parancsok, amik az előző eredményeket adták, ezek közül néhány feljebb is látható.

```
> D<-read.table("m2012.csv",sep=";",dec=".",header=TRUE)
> D$regio<-as.numeric(D$regio)
> M<-as.matrix(D[-c(1,2)])
> prcomp(M)
```


Standard deviations:

```
[1] 550073.986932887 138110.327987381 10903.485551094
[4] 4.564594712      3.178480823      1.521760201      1.063363286
```

Rotation:

	PC1	PC2	PC3
gdp.growth	-0.00000077431827839	-0.0000004928868878	0.00001341330000
gdp	-0.00108173989187416	-0.0327307164610078	-0.99946358427468
hly	0.00000004934621557	0.0000031554512870	-0.00018897138874
infmor	-0.00000019262157668	0.0000039712930286	0.00007102834227
life.exp	0.00000171643495560	-0.0000101479972816	-0.00018108802232
sulphur	0.44523724259554404	0.8949169906466821	-0.02978888339965
nitro	0.89541198765771379	-0.4450306505431759	0.01360486701921

	PC4	PC5	PC6
gdp.growth	-0.091435408052271	0.673998283258810	0.708544283882611
gdp	-0.000249043346568	0.000071736953363	-0.000081096780017
hly	0.914196243332683	0.342836146236879	-0.174218990326979
infmor	-0.065334747189896	0.250082968447074	-0.470185668326457
life.exp	0.389379125749417	-0.604688510483473	0.496526103929754
sulphur	-0.000006629774273	-0.000004212294638	0.000004403638392
nitro	0.000002105825321	0.000003958091492	-0.000002718275273

	PC7
gdp.growth	-0.187965097597959
gdp	0.000001447975012
hly	0.127891802705094
infmor	-0.843869250501463
life.exp	-0.486001540056652
sulphur	-0.000001586070212
nitro	0.000001370911598

```
> princomp(M)
```

Call:

```
princomp(x = M)
```

Standard deviations:

Comp.1	Comp.2	Comp.3	Comp.4
540161.932059879	135621.649769433	10707.010259302	4.482342990

Comp.5	Comp.6	Comp.7
3.121206200	1.494338854	1.044202042

7 variables and 28 observations.

```
> factanal(M,3)
```

Call:

```
factanal(x = M, factors = 3)
```

Uniquenesses:

	gdp.growth	gdp	hly	infmor	life.exp	sulphur	nitro
	0.005	0.497	0.717	0.499	0.005	0.005	0.237

Loadings:

	Factor1	Factor2	Factor3
gdp.growth	-0.119	-0.106	0.985
gdp	0.696	-0.130	
hly	0.516	0.113	
infmor	-0.664		0.245
life.exp	0.919	0.143	-0.361
sulphur	-0.122	0.987	
nitro	0.161	0.856	

	Factor1	Factor2	Factor3
SS loadings	2.090	1.770	1.175
Proportion Var	0.299	0.253	0.168
Cumulative Var	0.299	0.551	0.719

Test of the hypothesis that 3 factors are sufficient.

The chi square statistic is 6.58 on 3 degrees of freedom.

The p-value is 0.0864

```
# cel: D$regio
```

```
# magyarazo D[-(1:2)]
```

```
library(neuralnet)
```

```
library(ROCR)
```

```
> h<-24
```

```
> M<-neuralnet(regio~gdp.growth+gdp+hly+infmor+life.exp+sulphur+nitro,
               D,hidden=h)
```

```
> M
```

```
Call: neuralnet(formula = regio ~ gdp.growth + gdp + hly +
                infmor + life.exp + sulphur + nitro,
                data = D, hidden = h)
```

```
1 repetition was calculated.
```

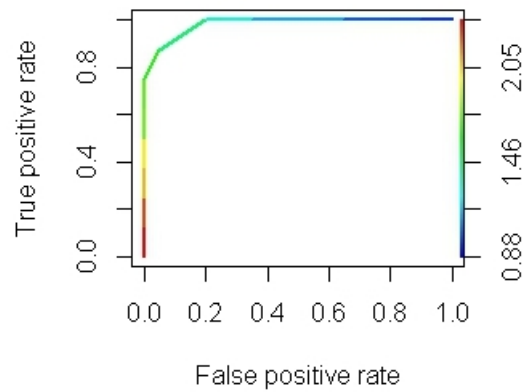
```
Error Reached Threshold Steps
```

```
1 1.274017018    0.009676005507 1371
```

```
> summary(M)
```

	Length	Class	Mode
call	4	-none-	call
response	28	-none-	numeric
covariate	196	-none-	numeric
model.list	2	-none-	list
err.fct	1	-none-	function
act.fct	1	-none-	function
linear.output	1	-none-	logical
data	9	data.frame	list
net.result	1	-none-	list
weights	1	-none-	list
startweights	1	-none-	list
generalized.weights	1	-none-	list
result.matrix	76	-none-	numeric

```
> P<-compute(M,D[-c(1,2)])
> pred <- prediction(P$net.res, D$regio)
> perf <- performance(pred,"tpr","fpr")
> plot(perf, colorize=TRUE)
```



Ez alapján a vágási szintet 1,5-re állítva:

```
> pred.num<-as.numeric(P$net.res<1.5)
> table(D$regio,pred.num)
```

```
pred.num
      0  1
1     0 20
2     3  5
```

8.2. Az adatbázis

Az adatok tehát az eurostat.com-ról származnak, 2012-ből. Az első oszlopban a vizsgált országok láthatók, a másodikban az általunk történt csoportosítás eredménye, a harmadikban a GDP növekedési ráta, utána a GDP, az ötödikben az születéskor várható egészségben töltött évek száma, a hatodikban a csecsemőhalandóság mértéke, a hetedikben a születéskor várható élettartam és végül a kibocsátott kén- és nitrogénoxidok millió tonnában.

	regio	gdp.gr rate	gdp	healty l.years	infmor	life exp	sulph.	nitro
Austria	a	0,9	34200	62,5	3,2	78,4	17899	187486
Belgium	a	0,1	31700	65,4	3,8	77,8	51611	216476
Bulgaria	b	0,5	11900	65,7	7,8	70,9	328815	126032
Cyprus	a	-2,4	24700	64	3,5	78,9	17273	22073
Denmark	a	-0,7	33100	61,4	3,4	78,1	18405	162565
Estonia	b	4,7	18800	57,2	3,6	71,4	45040	49591
Finland	a	-1,5	30500	56,2	2,4	77,7	54380	156089
France	a	0,3	28400	63,8	3,5	78,7	329300	1170009
Germany	a	0,4	32500	57,9	3,3	78,6	562121	1569702
Greece	a	-6,6	19600	64,9	2,9	78	391236	401809
Hungary	b	-1,5	17100	60,5	4,9	71,6	31836	122406
Iceland	a	1,1	30700	68	1,1	81,6	85539	27326
Ireland	a	-0,3	34300	68,5	3,5	78,7	24472	83065
Italy	a	-2,3	26900	61,5	2,9	79,8	283476	987656
Latvia	b	4,8	16000	59	6,3	68,9	7461	55638
Lithuania	b	3,8	18300	61,6	3,9	68,4	36399	57782
Luxembourg	a	-0,2	69800	66,4	2,5	79,1	2066	46404
Malta	a	2,5	22400	72,2	5,3	78,6	21809	19848
Netherlands	a	-1,6	35000	58,9	3,7	79,3	60734	354439
Norway	a	2,7	50300	70,4	2,5	79,5	21141	186915
Poland	a	1,8	17400	62,9	4,6	72,7	853913	824942
Portugal	a	-3,3	20200	62,6	3,4	77,3	73982	216521
Romania	b	0,6	14000	57,7	9	71	259666	225816
Slovakia	b	1,6	19600	53,1	5,8	72,5	58523	80990
Slovenia	b	-2,6	21600	55,6	1,6	77,1	10234	49428
Spain	a	-2,1	24900	65,8	3,1	79,5	1082838	1605671
Sweden	a	-0,3	33400	70,6	2,6	79,9	55205	254616
U. Kingdom	a	0,7	28400	64,5	4,1	79,1	514149	1403689

Irodalomjegyzék

- [1] Andrey Nikolaevich Kolmogorov: *On the representation of continuous functions of several variables as superposition of continuous functions of one variable and addition*, Dokl. Akad. Nauk SSSR, 114; 953-956, (1957) Translated in: Amer. Math. Soc. Transl. 28; 55-59 (1963)
- [2] David A. Sprecher: *A universal mapping for Kolmogorov's Superposition Theorem*, Neural Networks Volume 6, Issue 8; 1089-1094 (1993)
- [3] Robert Hecht-Nielsen: *Kolmogorov's Mapping Neural Network Existence Theorem*, First Annual International Conference on Neural Networks, Volume 3; 11-14, IEEE (1987)
- [4] Roman Neruda, Arnost Stedry, Jitka Drkosova: *Implementation remarks on learning algorithm based on Kolmogorov Theorem*, International Conference on Computational Science Volume 2; 986-995, Springer-Verlag London (2001)
- [5] Federico Girosi, Tomaso Poggio: *Representation properties of networks: Kolmogorov's Theorem is irrelevant*, Neural Computation Volume 1 Issue 4; 465-469, MIT Press Cambridge (1989)
- [6] George Cybenko: *Approximation by superpositions of a sigmoidal function*, Math. Control Signal Systems 2; 303-314, Springer-Verlag New York (1989)
- [7] Vera Kurkova: *Kolmogorov Theorem is relevant*, Neural Computation, Volume 3, No. 4; 617-622, Massachusetts Institute of Technology (1991)
- [8] Vasco Brattka: *From Hilbert's 13th Problem to the theory of neural networks: constructive aspects of Kolmogorov's Superposition Theorem*, Kolmogorov's Heritage in Mathematics; 253-280, Springer (2007)
- [9] Fazekas István: *Neurális hálózatok*, Debreceni Egyetem, egyetemi jegyzet (2013)
- [10] Frauke Günther, Stefan Fritsch: *Neuralnet: Trainig of Neural Network*, The R Journal 2/1; 30-38, (2010 June)

- [11] Christoph Bergmeir, José M. Benítez: *Neural Network in R Using the Stuttgart Neural Network Simulator: RSNNS*, Journal of Statistical Software 46/7; 1-26, American Statistical Association (2012 January)
- [12] Pavel Golik, Patrick Doetsch, Hermann Ney: *Cross-Entropy vs. Squared Error Training: a Theoretical and Experimental Comparison*, INTERSPEECH; 1756-1760 ISCA (2013)
- [13] Kyriaki Kitikidou, Lazaros Iliadis: *Developing Neural Networks to Investigate Relationships Between Air Quality and Quality of Life Indicators*, Air Pollution; 245-258 InTech (2011)
- [14] Walter Rudin: *Real and Complex Analysis*, McGraw-Hill (1966)
- [15] Walter Rudin: *Functional Analysis*, McGraw-Hill (1973)
- [16] Anatoli Georgievich Vitushkin: *On Hilbert's Thirteenth Problem*, Dokl. Akad. Nauk SSSR, 95; 701-704 (1954)
- [17] Anatoli Georgievich Vitushkin: *Some Properties of Linear Superposition of Smooth Function*, Dokl. Akad. Nauk SSSR 156; 1003-1006 (1964)
- [18] Anatoli Georgievich Vitushkin, Gennadi Markovich Henkin: *Linear Superposition of Functions*, Russian Math. Surveys 22; 77-125 (1967)
- [19] S. L. Özesmi, U. Özesmi: *An Artificial Neural Network Approach to Spatial Habitat Modeling with Interspecific Interaction*, Ecological Modelling 116; 15-31 (1999)