Alkalmazott Matematikai Lapok **24** (2007), 209-222.

TÖBBRÉTEGŰ KERCEPTRON

SZABÓ ZOLTÁN, LŐRINCZ ANDRÁS

Többrétegű Perceptronokba (MLP) Támasztó Vektor Gépeket (SVM) ágyazva többrétegű SVM hálókat konstruálunk. Az összekapcsolt approximációs forma az SVM-ek általánosító képességét és az MLP-k rejtett rétegéből adódó kombinatorikus tulajdonságot egyaránt kihasználhatja. A hálózatot Többrétegű Kerceptron (MLK) hálózatnak nevezzük. Az MLK rendelkezik hibavisszaterjesztésen alapuló hangolási eljárással, amit jelen munkában bemutatunk. Négyzetes költségfüggvényre – regularizációs lehetőségekkel – hangolási szabályt származtatunk. Megközelítésünk egy további tulajdonsága, hogy az ún. *kernel trükk* segítségével az MLK-hoz tartozó számítások a duális térben kivitelezhetők.

1. Bevezetés

A Többrétegű Perceptronokat (MLP) és a Támasztó Vektor Gépeket (SVM) széles körben tanulmányozták az irodalomban. Kiváló áttekintést ad a témában [1, 8]. Munkánkban az SVM-eket többrétegű formára terjesztjük ki, és a kapott rendszerre hibavisszaterjesztésen alapuló hangolási szabályt vezetünk le. Az ún. kernel trükk alkalmazásával, a problémát skaláris szorzat segítségével tárgyaljuk. Más, ugyanezt a trükköt használó eljárások leírása megtalálható a [5, 10, 11] hivatkozásokban.

2. A hálózat felépítése

2.1. Jelölések

Különböző betűtípussal jelöljük a számokat (a), a vektorokat ($\mathbf{a}$), és a mátrixokat ($\mathbf{A}$). $\mathbf{A}^T$ az $\mathbf{A}$ mátrix transzponáltja. Az $\mathbf{a}$ vektor egy a komponenssel való kibővítését $[\mathbf{a}; a]$-ként írjuk. $\mathbb{R}$ szimbolizálja a valós számokat. $\|\cdot\|_2$ jelöli az E Euklideszi térbeli $\langle\cdot,\cdot\rangle$ skaláris szorzat által indukált L_2 normát, azaz $\|\mathbf{e}\|_2 = \sqrt{\langle \mathbf{e}, \mathbf{e}\rangle}$ $(\mathbf{e} \in E)$.

2.2. Építőelemek

2.2.1. SVM

Az SVM-ek gyakran használt approximációs eszközök [3, 4, 9, 10, 6]. $\{\mathbf{x}(t), d(t)\}_{t=1..T}$ mintapárokat közelítenek, ahol az $\mathbf{x}(t)$ input az $\mathcal{X}$ *input térből* származik, és $d(t) \in \mathbb{R}$. A közelítés lineáris egy alkalmas $\mathcal{H}$ térben. Ebbe a térbe a

$$\boldsymbol{\varphi} : \mathbf{x} \in \mathcal{X} \to \mathcal{H}$$

hozzárendelés képezi le az $\mathbf{x}(t)$ inputokat. $\boldsymbol{\varphi}(\mathbf{x})$-et az $\mathbf{x}$ input *reprezentációjaként* interpretálhatjuk. Az SVM közelítés a

$$f_{\mathbf{w}} : \mathbf{x} \in \mathcal{X} \mapsto \langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{x}) \rangle_{\mathcal{H}} \quad (\mathbf{w} \in \mathcal{H})$$

formájú. Formálisan, az SVM feladat

$$\min_{\mathbf{w}} H[\mathbf{w}] := C \cdot \sum_{t=1}^{T} V[d(t), f_{\mathbf{w}}(\mathbf{x}(t))] + \frac{1}{2} \|\mathbf{w}\|_{\mathcal{H}}^2 \quad (C > 0),$$

ahol $V[\cdot, \cdot]$ az ún. *veszteségfüggény*, amely lehet kvadratikus, ϵ-érzéketlen, de más formákat is szoktak használni [5]. Röviden, az SVM-ek regularizált lineáris approximátorok [8].

Az explicit $\boldsymbol{\varphi}$ leképezés helyett a $\mathcal{H}$ tér egy k kernel segítségével is leírható, $\mathcal{H} = \mathcal{H}(k)$ [7], ahol $\boldsymbol{\varphi}(\mathbf{x}) = k(\cdot, \mathbf{x})$. A k kernel a

$$\langle f(\cdot), k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}} = f(\mathbf{x}) \quad (\mathbf{x} \in \mathcal{X}, \forall f \in \mathcal{H}),$$

reprodukáló tulajdonsággal rendelkezik [2, 7], és $\mathcal{H}$-t Reprodukáló Kernel Hilbert Térnek (RKHS) nevezzük. Tehát, tetszőleges $f \in \mathcal{H}$ RKHS-beli függvény $k(\cdot, \mathbf{x})$ kernellel való skaláris szorzata az $\mathbf{x}$ pontbeli kiértékelésnek felel meg. A skaláris szorzat a $\mathcal{H}$ térben implicit módon számolható a kernel segítségével

$$k(\mathbf{u}, \mathbf{v}) = \langle \boldsymbol{\varphi}(\mathbf{u}), \boldsymbol{\varphi}(\mathbf{v}) \rangle_{\mathcal{H}} \quad (\mathbf{u}, \mathbf{v} \in \mathcal{X}).$$

Így a $\mathbf{w} = \sum_{j=1}^{N} \alpha_j \cdot \boldsymbol{\varphi}(\mathbf{z}_j) \quad (\alpha_j \in \mathbb{R}, \mathbf{z}_j \in \mathcal{X})$ választással

$$f_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{x}) \rangle_{\mathcal{H}} = \sum_{j=1}^{N} \alpha_j \cdot \langle \boldsymbol{\varphi}(\mathbf{z}_j), \boldsymbol{\varphi}(\mathbf{x}) \rangle_{\mathcal{H}} = \sum_{j=1}^{N} \alpha_j \cdot k(\mathbf{z}_j, \mathbf{x}).$$

Tehát az $f_{\mathbf{w}}$ függvény az α_j együtthatók, a $\mathbf{z}_j$ minták és a k kernel segítségével a $\boldsymbol{\varphi}(\mathbf{x})$ reprezentáció explicit felhasználása nélkül kiértékelhető. Ez a fogás a *kernel trükk*.

2.2.2. MLP

Az MLP neurális hálózat többrétegű, minden réteg egy

$$\mathbf{x} \mapsto \mathbf{g}(\mathbf{W} \cdot \mathbf{x}) \tag{1}$$

formájú nemlineáris leképezést valósít meg. Itt $\mathbf{g}$ egy differenciálható nemlineáris függvény. Az MLP feladatban úgy hangoljuk az egyes rétegek $\mathbf{W}$ mátrixait, hogy a hálózat az $\{\mathbf{x}(t), \mathbf{d}(t)\}$ input-output minta pároknak megfelelő leképezést közelítse. Formálisan, célunk a

$$\varepsilon^2(t) := \|\mathbf{d}(t) - \mathbf{y}(t)\|_2^2 \to \min_{\mathbf{W}_1, \mathbf{W}_2, \ldots}, \tag{2}$$

kvadratikus hiba minimalizálása, ahol $\mathbf{y}(t)$ jelöli a hálózat t időpontbeli kimenetét. Az MLP feladatot oldja meg a jól ismert *visszaterjesztési algoritmus*.

2.3. Az MLK hálózat

Egy általános MLP réteg által megvalósított leképezés [lásd az (1) egyenletet]

$$\mathbf{x} \mapsto \mathbf{g}\left(\begin{bmatrix} \vdots \\ \langle \mathbf{w}_i, \mathbf{x} \rangle \\ \vdots \end{bmatrix}\right)$$

formájú, ahol $\mathbf{w}_i^T$ a $\mathbf{W}$ mátrix i. sorát jelöli. SVM illeszthető az MLP-be, ha a hálózat egy általános rétege a

$$\mathbf{x} \mapsto \mathbf{g}\left(\begin{bmatrix} \langle \mathbf{w}_1, \boldsymbol{\varphi}(\mathbf{x}) \rangle_{\mathcal{H}} \\ \vdots \\ \langle \mathbf{w}_N, \boldsymbol{\varphi}(\mathbf{x}) \rangle_{\mathcal{H}} \end{bmatrix}\right)$$

hozzárendelést valósítja meg.[1]

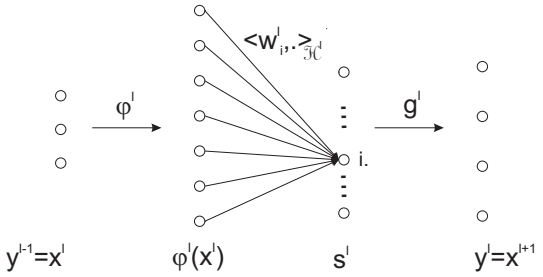
Egy ilyen rétegekből felépített hálózatot Többrétegű Kerceptronnak (MLK) fogunk hívni, lásd az 1. ábrát.

Minden egyes réteg ($\mathbf{x}^l$) inputját az előző réteg ($\mathbf{y}^{l-1}$) outputja szolgáltatja. A 0-adik réteg a külvilág, ami az MLK első rétege számára szolgáltatja a bemenetet.

$$\mathbf{x}^l = \mathbf{y}^{l-1} \in \mathbb{R}^{N_{\mathrm{I}}^l},$$

ahol N_{I}^l az l-edik réteg dimenziója. Az l-edik réteg $\mathbf{x}^l$ inputja a $\boldsymbol{\varphi}^l$ leképezésen átesve a $\mathbf{w}_i^l$ súlyokkal szorzódik. Ez a két lépcsős eljárás implicit módon elvégezhető a k^l kerneleket és a $\mathbf{w}_i^l$-k kifejtéseit használva. Az adódó $\mathbf{s}^l \in \mathbb{R}^{N_{\mathrm{S}}^l}$ vektorra hat

[1] Az egyszerűség kedvéért válasszuk az $\mathcal{X}$ input teret a véges dimenziós Euklideszi-térnek, azaz $\mathbb{R}^n$-nek.



1. ábra. Az MLK hálózat l-edik rétege, $l = 1, 2, \ldots L$. Minden egyes réteg inputját ($\mathbf{x}^l$) az előző réteg outputja adja ($\mathbf{y}^{l-1}$). A 0-adik réteg a külvilág, ami az MLK első réteg számára szolgáltatja a bemenetet. Az l-edik réteg $\mathbf{x}^l$ inputja a $\boldsymbol{\varphi}^l$ leképezésen esik át, majd a réteg $\mathbf{w}_i^l$ súlyaival szorzódik skalárisan a $\mathcal{H}^l = \mathcal{H}^l(k^l)$ RKHS-ben. Az adódó $\mathbf{s}^l$ vektorra hat a $\mathbf{g}^l$ differenciálható nemlinearitás. Ezen nemlineáris függvény kimenete a következő réteg bemenete, $\mathbf{x}^{l+1}$. A hálózat kimenete az utolsó réteg kimenete.

a $\mathbf{g}^l$ nemlineáris, differenciálható függvény. Ennek a nemlineáris függvénynek a kimenete a következő réteg bemenete, azaz $\mathbf{x}^{l+1}$. Az utolsó (L.) réteg outputját – azaz a hálózat outputját – $\mathbf{y}$ jelöli.

$$\mathbf{y}^l = \mathbf{x}^{l+1} \in \mathbb{R}^{N_{\mathrm{o}}^l},$$

és az l-edik réteg kimenetének dimenziója N_{o}^l.

A következőkben megmutatjuk, hogy (i) az MLK-k is rendelkeznek visszaterjesztési szabállyal, ami (ii) csak kernelek segítségével is megadható, és így a számítások a duális térben kivitelezhetőek.

3. Az MLK visszaterjesztési eljárás

Egy kicsit általánosabb, regularizációs tagokat is tartalmazó feladat a

$$c(t) := \varepsilon^2(t) + r(t) \longrightarrow \min_{\{\mathcal{H}^l \ni \mathbf{w}_i^l:\ l=1,\ldots,L;\ i=1,\ldots,N_{\mathrm{S}}^l\}},$$

probléma, ahol

$$\varepsilon^2(t) = \|\mathbf{d}(t) - \mathbf{y}(t)\|_2^2, \quad r(t) = \sum_{l=1}^{L} \sum_{i=1}^{N_{\mathrm{S}}^l} \lambda_i^l \cdot \left\|\mathbf{w}_i^l(t)\right\|_{\mathcal{H}^l}^2 \qquad (\lambda_i^l \geq 0)$$

a költségfüggvény approximációs és regularizációs tagjai, és $\mathbf{y}(t)$ jelöli a hálózat t-edik inputra adott kimenetét. A λ_i^l paraméterek szabályozzák az approximáció és regularizáció közötti arányt. $\lambda_i^l = 0$-ra a legjobb közelítést keressük, mint az MLP feladatban [(2) egyenlet]. λ_i^l értékeket növelve az approximáció simasága nő.

A fenti jelölésekkel a következő állítások igazolhatók.

3.1. TÉTEL. (explicit eset) *Tegyük fel, hogy az* $\mathbf{x} \mapsto \langle \mathbf{w}, \boldsymbol{\varphi}^l(\mathbf{x}) \rangle_{\mathcal{H}^l}$ *és a* $\mathbf{g}^l$ *függvények differenciálhatók* $(l = 1, \ldots, L)$. *Ekkor visszaterjesztési szabály származtatható az MLK-ra, ha a költségfüggvény*

$$c(t) = \varepsilon^2(t) + \sum_{l=1}^{L} \sum_{i=1}^{N_{\mathrm{S}}^l} \lambda_i^l \cdot \left\| \mathbf{w}_i^l(t) \right\|_{\mathcal{H}^l}^2 \quad (\lambda_i^l \geq 0) \text{ alakú.}$$

3.2. TÉTEL. (implicit eset) *Tegyük fel, hogy az alábbiak teljesülnek:*

1. *Differenciálhatósági megkötés: A* k^l *kernelek mindkét változójukban, illetve a* $\mathbf{g}^l$ *függvények differenciálhatók* $(l = 1, \ldots, L)$.
2. *Kifejtési tulajdonság: A hálózat kezdeti* $\mathbf{w}_i^l(1)$ *súlyai egy adott*

$$\mathcal{H}^l \ni \mathbf{w}_i^l(1) = \sum_{j=1}^{N_i^l(1)} \alpha_{i,j}^l(1) \cdot \boldsymbol{\varphi}^l(\mathbf{z}_{i,j}^l(1)) \quad (l = 1, \ldots, L; i = 1, \ldots, N_{\mathrm{S}}^l) \quad (3)$$

típusú kifejtéssel, duális reprezentációval rendelkeznek.

Ekkor létezik visszaterjesztési eljárás az MLK hálózatra, feltéve, hogy a költségfüggvény

$$c(t) = \varepsilon^2(t) + \sum_{l=1}^{L} \sum_{i=1}^{N_{\mathrm{S}}^l} \lambda_i^l \cdot \left\| \mathbf{w}_i^l(t) \right\|_{\mathcal{H}^l}^2 \quad (\lambda_i^l \geq 0)$$

formájú. Az eljárás megőrzi a (3) tulajdonságot, ami így a behangolt hálózatra is fennáll. Az algoritmus implicit abban az értelemben, hogy a duális térben realizálható.

Az MLK visszaterjesztési eljárások pszeudokódjai a 3.1. és a 3.2. táblázatban találhatók. Az algoritmusok levezetését, mind az explicit mind az implicit esetre a következő alfejezetben adjuk meg.

Az MLK visszaterjesztési eljárások szemléletesen (párhuzamosan lásd a 3.1. és a 3.2. táblázatokat):

1. A $\boldsymbol{\delta}^l(t)$ visszaterjesztett hiba $\boldsymbol{\delta}^L(t)$-ből indulva egy hátráló rekurzióval fejlődik a $\frac{d[\mathbf{s}^{l+1}(t)]}{d[\mathbf{s}^l(t)]}$ deriválton keresztül.

2. A $\frac{d[\mathbf{s}^{l+1}(t)]}{d[\mathbf{s}^l(t)]}$ kifejezés a $\boldsymbol{\varphi}^{l+1}$ leképezés, vagy implicit módon a k^{l+1} kernel segítségével határozható meg.

3. $\mathbf{w}$-k hangolásában két tényező játszik szerepet:

   (a) *Felejtés* valósul meg a $\mathbf{w}_i^l$ súlyok $\left(1 - 2\mu_i^l(t) \cdot \lambda_i^l\right)$-szeres szorzása által, ahol λ_i^l a regularizációs együttható.

(b) *Adaptáció* jelenik meg a visszaterjesztett hibán keresztül. Az l-edik réteg súlyait az $\mathbf{x}^l(t)$ reprezentáció, azaz az aktuális inputnak az l-edik rétegre leképezett értéke állítja úgy, hogy a hangolást a visszaterjesztett hiba súlyozza.

3.1. táblázat. (Az explicit MLK visszaterjesztési algoritmus pszeudokódja.)

Algoritmus bemenete
mintapontok: $\{\mathbf{x}(t), \mathbf{d}(t)\}_{t=1,\dots,T}, T$
költségfüggvény: $\lambda_i^l \geq 0$ $(l = 1, \dots, L; i = 1, \dots, N_{\mathrm{S}}^l)$
tanulási ráták: $\mu_i^l(t) > 0$ $(l = 1, \dots, L; i = 1, \dots, N_{\mathrm{S}}^l; t = 1, \dots, T)$
Hálózat inicializációja
méretek: L (rétegek száma), N_{I}^l, N_{S}^l, N_{o}^l $(l = 1, \dots, L)$
súlyok: $\mathbf{w}_i^l(1)$ $(l = 1, \dots, L; i = 1, \dots, N_{\mathrm{S}}^l)$
Számítás kezdete
Aktuális input $\mathbf{x}(t)$
Előreterjesztés
$\mathbf{x}^l(t)$ $(l = 2, \dots, L+1)$, $\mathbf{s}^l(t)$ $(l = 2, \dots, L)$[2]
Hiba visszaterjesztése
$l = L$
while $l \geq 1$
if $(l = L)$
$\boldsymbol{\delta}^L(t) = 2 \cdot [\mathbf{y}(t) - \mathbf{d}(t)]^T \cdot \left(\mathbf{g}^L\right)' (\mathbf{s}^L(t))$
else

$$\frac{d[\mathbf{s}^{l+1}(t)]}{d[\mathbf{s}^l(t)]} = \begin{bmatrix} \vdots \\ \left.\frac{d\left[\left\langle \mathbf{w}_i^{l+1}(t), \boldsymbol{\varphi}^{l+1}(\mathbf{u})\right\rangle_{\mathcal{H}^{l+1}}\right]}{d[\mathbf{u}]}\right|_{\mathbf{u}=\mathbf{x}^{l+1}(t)} \\ \vdots \end{bmatrix} \cdot \left[\left(\mathbf{g}^l\right)' (\mathbf{s}^l(t))\right]^{3}$$

$\boldsymbol{\delta}^l(t) = \boldsymbol{\delta}^{l+1}(t) \cdot \frac{d[\mathbf{s}^{l+1}(t)]}{d[\mathbf{s}^l(t)]}$
Súlyok frissítése
for $\forall\, i$: $1 \leq i \leq N_S^l$
$\mathbf{w}_i^l(t+1) = (1 - 2\mu_i^l(t) \cdot \lambda_i^l) \cdot \mathbf{w}_i^l(t) - \mu_i^l(t) \cdot \delta_i^l(t) \cdot \boldsymbol{\varphi}^l(\mathbf{x}^l(t))$
$l = l - 1$
Számítás vége

[2] Így a hálózat kimenete, azaz $\mathbf{y}(t) = \mathbf{x}^{L+1}(t)$ is kiszámítódik.
[3] Itt: $i = 1, \dots, N_{\mathrm{S}}^{l+1}$.

3.2. táblázat. (Az implicit MLK visszaterjesztési algoritmus pszeudokódja.)

Algoritmus bemenete
mintapontok: $\{\mathbf{x}(t), \mathbf{d}(t)\}_{t=1,\dots,T}, T$
költségfüggvény: $\lambda_i^l \geq 0$ $(l = 1, \dots, L; i = 1, \dots, N_{\mathrm{S}}^l)$
tanulási ráták: $\mu_i^l(t) > 0$ $(l = 1, \dots, L; i = 1, \dots, N_{\mathrm{S}}^l; t = 1, \dots, T)$
Hálózat inicializációja
méretek: L (rétegek száma), N_{I}^l, N_{S}^l, N_{o}^l $(l = 1, \dots, L)$
együtthatók: $\boldsymbol{\alpha}_i^l(1) \in \mathbb{R}^{N_i^l(1)}$
ősök: $\mathbf{z}_{i,j}^l(1)$, ahol $j = 1, \dots, N_i^l(1)$
Számítás kezdete
Aktuális input $\mathbf{x}(t)$
Előreterjesztés
$\mathbf{x}^l(t)$ $(l = 2, \dots, L+1)$, $\mathbf{s}^l(t)$ $(l = 2, \dots, L)$[4]
$l = L$
while $l \geq 1$
if $(l = L)$
$\boldsymbol{\delta}^L(t) = 2 \cdot [\mathbf{y}(t) - \mathbf{d}(t)]^T \cdot \left(\mathbf{g}^L\right)' (\mathbf{s}^L(t))$
else

$$\frac{d[\mathbf{s}^{l+1}(t)]}{d[\mathbf{s}^l(t)]} = \begin{bmatrix} & \vdots & \\ \sum_{j=1}^{N_i^{l+1}(t)} \alpha_{ij}^{l+1}(t) \cdot [k^{l+1}]'_y(\mathbf{z}_{ij}^{l+1}(t), \mathbf{x}^{l+1}(t)) & \\ & \vdots & \end{bmatrix} \cdot \left[\left(\mathbf{g}^l\right)' (\mathbf{s}^l(t))\right]^5$$

$\boldsymbol{\delta}^l(t) = \boldsymbol{\delta}^{l+1}(t) \cdot \frac{d[\mathbf{s}^{l+1}(t)]}{d[\mathbf{s}^l(t)]}$
Súlyok frissítése
for $\forall\, i$: $1 \leq i \leq N_S^l$
$N_i^l(t+1) = N_i^l(t) + 1$
$\boldsymbol{\alpha}_i^l(t+1) = \left[\left(1 - 2\mu_i^l(t) \cdot \lambda_i^l\right) \cdot \boldsymbol{\alpha}_i^l(t); -\mu_i^l(t) \cdot \delta_i^l(t)\right]$
$\mathbf{z}_{i,j}^l(t+1) = \mathbf{z}_{i,j}^l(t) \quad (j = 1, \dots, N_i^l(t))$
$\mathbf{z}_{i,j}^l(t+1) = \mathbf{x}^l(t) \quad (j = N_i^l(t+1))$
$l = l - 1$
Számítás vége

[4] Így a hálózat kimenete, azaz $\mathbf{y}(t) = \mathbf{x}^{L+1}(t)$ is kiszámítódik.
[5] $i = 1, \dots, N_{\mathrm{S}}^{l+1}$

3.1. Az MLK visszaterjesztési eljárások levezetése

Először a $\frac{d[c(t)]}{d[\mathbf{w}_i^l(t)]}$ gradienst származtatjuk. Utána a gradienst a legmeredekebb lejtő módszerbe ágyazzuk.[6] A $c(t)$ hiba két tagból áll, approximációs és regularizációs tagból:

$$c(t) = \varepsilon^2(t) + r(t).$$

3.1.1. Az approximációs tag gradiense

Először néhány MLK felépítéséből adódó alapösszefüggést sorolunk fel. Az egyszerűség kedvéért a továbbiakban a t indexet elhagyjuk [precízen: $\mathbf{x}^l = \mathbf{x}^l(t)$, $\mathbf{y}^l = \mathbf{y}^l(t)$, $\mathbf{s}^l = \mathbf{s}^l(t)$,$\mathbf{w}_i^l = \mathbf{w}_i^l(t)$].

$$\begin{aligned}
\mathbf{x}^l &= \mathbf{y}^{l-1} \in \mathbb{R}^{N_I^l} \qquad (l = 1, \ldots, L+1) \\
\mathbf{x}^{l+1} &= \mathbf{g}^l(\mathbf{s}^l) \qquad (l = 1, \ldots, L) \qquad (4) \\
\mathbf{s}^l &= \begin{bmatrix} \langle \mathbf{w}_1^l, \boldsymbol{\varphi}^l(\mathbf{x}^l) \rangle_{\mathcal{H}^l} \\ \vdots \\ \langle \mathbf{w}_i^l, \boldsymbol{\varphi}^l(\mathbf{x}^l) \rangle_{\mathcal{H}^l} \\ \vdots \end{bmatrix} \qquad (l = 1, \ldots, L; i = 1, \ldots, N_{\mathrm{S}}^l) \\
&= \begin{bmatrix} \langle \mathbf{w}_1^l, \boldsymbol{\varphi}^l(\mathbf{g}^{l-1}(\mathbf{s}^{l-1})) \rangle_{\mathcal{H}^l} \\ \vdots \\ \langle \mathbf{w}_i^l, \boldsymbol{\varphi}^l(\mathbf{g}^{l-1}(\mathbf{s}^{l-1})) \rangle_{\mathcal{H}^l} \\ \vdots \end{bmatrix} \qquad (l = 2, \ldots, L; i = 1, \ldots, N_{\mathrm{S}}^l) \\
\mathbf{s}^{l+1} &= \begin{bmatrix} \langle \mathbf{w}_1^{l+1}, \boldsymbol{\varphi}^{l+1}(\mathbf{g}^l(\mathbf{s}^l)) \rangle_{\mathcal{H}^{l+1}} \\ \vdots \\ \langle \mathbf{w}_i^{l+1}, \boldsymbol{\varphi}^{l+1}(\mathbf{g}^l(\mathbf{s}^l)) \rangle_{\mathcal{H}^{l+1}} \\ \vdots \end{bmatrix} \qquad (l = 1, \ldots, L-1; i = 1, \ldots, N_{\mathrm{S}}^{l+1}) \qquad (5)
\end{aligned}$$

Az l-edik réteg visszaterjesztett hibáját definiáljuk a

$$\boldsymbol{\delta}^l(t) := \frac{d[\varepsilon^2(t)]}{d[\mathbf{s}^l(t)]} \quad (l = 1, \ldots, L)$$

[6] A legmeredekebb lejtő módszerét használjuk ötletünk bemutatásához. Más, ettől eltérő gradiens alapú technikák szintén szóba jöhetnének. Például a momentum módszer, illetve a konjugált gradiens eljárások is rendelkeznek előnyös tulajdonságokkal.

módon. Speciálisan, az utolsó rétegre:

$$\begin{aligned}\boldsymbol{\delta}^L(t) = \frac{d[\varepsilon^2(t)]}{d[\mathbf{s}^L(t)]} &= \frac{d\left[\left\|\mathbf{d}(t) - \mathbf{g}^L(\mathbf{s}^L(t))\right\|_2^2\right]}{d[\mathbf{s}^L(t)]} \\ &= 2 \cdot \left[\mathbf{g}^L\left(\mathbf{s}^L(t)\right) - \mathbf{d}(t)\right]^T \cdot \left(\mathbf{g}^L\right)'\left(\mathbf{s}^L(t)\right) \\ &= 2 \cdot \left[\mathbf{y}(t) - \mathbf{d}(t)\right]^T \cdot \left(\mathbf{g}^L\right)'\left(\mathbf{s}^L(t)\right).\end{aligned}$$

Itt először a láncszabályt, majd a vektorokra érvényes

$$\frac{d[\|\mathbf{d} - \mathbf{y}\|_2^2]}{d\mathbf{y}} = 2(\mathbf{y} - \mathbf{d})^T$$

összefüggést használtuk ki, végül beillesztettük az MLK szerkezetéből adódó

$$\mathbf{y}(t) = \mathbf{g}^L\left(\mathbf{s}^L(t)\right)$$

azonosságot.

A

$$\frac{d[\mathbf{s}^{l+1}(t)]}{d[\mathbf{s}^l(t)]} \quad (l = 1, \ldots, L-1) \tag{6}$$

kifejezés az (5) egyenlet segítségével számolható. Elégséges a

$$\frac{d[\langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{g}(\mathbf{s}))\rangle_{\mathcal{H}}]}{d[\mathbf{s}]} \tag{7}$$

alakú kifejezéseket tekintenünk, abból a teljes derivált „kirakható". (7) értéke az alábbi lemma alkalmazásával megadható.

3.1. Lemma. *Legyen $\mathbf{w} \in \mathcal{H} = \mathcal{H}(k)$ egy RKHS-beli pont. Tegyük fel, hogy:*

1. *A k kernel mindkét argumentuma szerint differenciálható, és jelölje k'_y a kernel második argumentuma szerint vett deriváltját.*
2. *Implicit esetben feltételezzük még, hogy $\mathbf{w}$ véges sok $\mathbf{z}_i$ pont $\mathcal{H}$-beli reprezentációjának képterében fekszik. Azaz*

$$\mathbf{w} \in Im\left(\boldsymbol{\varphi}(\mathbf{z}_1), \boldsymbol{\varphi}(\mathbf{z}_2), \ldots, \boldsymbol{\varphi}(\mathbf{z}_N)\right) \subseteq \mathcal{H}.$$

Legyen ez a kifejtés $\mathbf{w} = \sum_{j=1}^{N} \alpha_j \cdot \boldsymbol{\varphi}(\mathbf{z}_j)$, ahol $\alpha_j \in \mathbb{R}$.

Ekkor:

1. *Explicit eset:*

$$\frac{d[\langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{g}(\mathbf{s}))\rangle_{\mathcal{H}}]}{d[\mathbf{s}]} = \left.\frac{d\left[\langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{u})\rangle_{\mathcal{H}}\right]}{d[\mathbf{u}]}\right|_{\mathbf{u}=\mathbf{g}(\mathbf{s})} \cdot \mathbf{g}'(\mathbf{s})$$

2. *Implicit eset:*

$$\frac{d[\langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{g}(\mathbf{s}))\rangle_{\mathcal{H}}]}{d[\mathbf{s}]} = \sum_{j=1}^{N} \alpha_j \cdot k'_y(\mathbf{z}_j, \mathbf{g}(\mathbf{s})) \cdot \mathbf{g}'(\mathbf{s})$$

Bizonyítás.

1. Explicit eset: az állítás adódik a láncszabályból.
2. Implicit eset:

$$\begin{aligned}
\frac{d[\langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{g}(\mathbf{s}))\rangle_{\mathcal{H}}]}{d[\mathbf{s}]} &= \frac{d\left[\left\langle \sum_j \alpha_j \cdot \boldsymbol{\varphi}(\mathbf{z}_j), \boldsymbol{\varphi}(\mathbf{g}(\mathbf{s}))\right\rangle_{\mathcal{H}}\right]}{d[\mathbf{s}]} \\
&= \frac{d\left[\sum_j \alpha_j \cdot \langle \boldsymbol{\varphi}(\mathbf{z}_j), \boldsymbol{\varphi}(\mathbf{g}(\mathbf{s}))\rangle_{\mathcal{H}}\right]}{d[\mathbf{s}]} \\
&= \frac{d\left[\sum_j \alpha_j \cdot k\left(\mathbf{z}_j, \mathbf{g}(\mathbf{s})\right)\right]}{d[\mathbf{s}]} \\
&= \sum_j \alpha_j \cdot k'_y(\mathbf{z}_j, \mathbf{g}(\mathbf{s})) \cdot \mathbf{g}'(\mathbf{s}).
\end{aligned}$$

Az első egyenletben beírtuk $\mathbf{w}$ kifejtését, majd kihasználtuk a skaláris szorzat linearitását. Ezután a

$$k(\mathbf{u}, \mathbf{v}) = \langle \boldsymbol{\varphi}(\mathbf{u}), \boldsymbol{\varphi}(\mathbf{v})\rangle_{\mathcal{H}}$$

reprezentáció és kernel közti összefüggést alkalmaztuk. Az utolsó lépés a láncszabályból adódik.

□

Folytatjuk (6) kiszámítását:

1. Explicit eset: Az előző lemma szerint

$$\begin{aligned}
\frac{d[\mathbf{s}^{l+1}(t)]}{d[\mathbf{s}^{l}(t)]} &= \begin{bmatrix} \vdots \\ \left.\frac{d\left[\left\langle \mathbf{w}_i^{l+1}(t), \boldsymbol{\varphi}^{l+1}(\mathbf{u})\right\rangle_{\mathcal{H}^{l+1}}\right]}{d[\mathbf{u}]}\right|_{\mathbf{u}=\mathbf{g}^l(\mathbf{s}^l(t))} \cdot \left(\mathbf{g}^l\right)'\left(\mathbf{s}^l(t)\right) \\ \vdots \end{bmatrix} \\
&= \begin{bmatrix} \vdots \\ \left.\frac{d\left[\left\langle \mathbf{w}_i^{l+1}(t), \boldsymbol{\varphi}^{l+1}(\mathbf{u})\right\rangle_{\mathcal{H}^{l+1}}\right]}{d[\mathbf{u}]}\right|_{\mathbf{u}=\mathbf{x}^{l+1}(t)} \\ \vdots \end{bmatrix} \cdot \left[\left(\mathbf{g}^l\right)'\left(\mathbf{s}^l(t)\right)\right] \\
&\qquad (l = 1, \ldots, L-1; i = 1, \ldots, N_{\mathrm{S}}^{l+1}).
\end{aligned}$$

A második egyenlőségnél (i) kihasználtuk a (4) azonosságot, és (ii) kiemeltük a $\left(\mathbf{g}^l\right)'\left(\mathbf{s}^l(t)\right)$ tagot mátrixok szorzásának megfelelően.

2. Implicit eset: $\mathbf{w}_i^{l+1}(t)$-kre fennáll a (3) kifejtési tulajdonság. Ez kezdetben feltevésünk volt. A 3.1.3. alfejezetben látni fogjuk, hogy ez a tulajdonság az iterációk során „öröklődik". Így

$$\mathbf{w}_i^{l+1}(t) = \sum_{j=1}^{N_i^{l+1}(t)} \alpha_{ij}^{l+1}(t) \cdot \boldsymbol{\varphi}^{l+1}(\mathbf{z}_{ij}^{l+1}(t)) \quad (l = 1, \ldots, L-1; i = 1, \ldots, N_{\mathrm{S}}^{l+1})$$

és a kívánt (6) derivált a lemma alkalmazásával

$$\frac{d[\mathbf{s}^{l+1}(t)]}{d[\mathbf{s}^l(t)]} = \begin{bmatrix} \vdots \\ \sum\limits_{j=1}^{N_i^{l+1}(t)} \alpha_{ij}^{l+1}(t) \cdot [k^{l+1}]'_y(\mathbf{z}_{ij}^{l+1}(t), \mathbf{g}^l(\mathbf{s}^l(t))) \cdot \left(\mathbf{g}^l\right)'(\mathbf{s}^l(t)) \\ \vdots \end{bmatrix} =$$

$$= \begin{bmatrix} \vdots \\ \sum\limits_{j=1}^{N_i^{l+1}(t)} \alpha_{ij}^{l+1}(t) \cdot [k^{l+1}]'_y(\mathbf{z}_{ij}^{l+1}(t), \mathbf{x}^{l+1}(t)) \\ \vdots \end{bmatrix} \cdot \left[\left(\mathbf{g}^l\right)'(\mathbf{s}^l(t))\right]$$

$(l = 1, \ldots, L-1; i = 1, \ldots, N_{\mathrm{S}}^{l+1})$.

A második egyenlőségnél kihasználtuk a (4) azonosságot. A $\left(\mathbf{g}^l\right)'\left(\mathbf{s}^l(t)\right)$ mátrix tagot mátrixok szorzásának megfelelően kiemeltük.

Láncszabály és $\boldsymbol{\delta}^{l+1}(t)$ definíciója alapján

$$\boldsymbol{\delta}^l(t) = \frac{d[\varepsilon^2(t)]}{d[\mathbf{s}^l(t)]} = \frac{d[\varepsilon^2(t)]}{d[\mathbf{s}^{l+1}(t)]} \cdot \frac{d[\mathbf{s}^{l+1}(t)]}{d[\mathbf{s}^l(t)]} = \boldsymbol{\delta}^{l+1}(t) \cdot \frac{d[\mathbf{s}^{l+1}(t)]}{d[\mathbf{s}^l(t)]} \quad (l = 1, \ldots, L-1).$$

Ismét alkalmazva a láncszabályt, $\boldsymbol{\delta}^l(t)$ és $\mathbf{s}^l(t)$ definíciója szerint

$$\frac{d[\varepsilon^2(t)]}{d[\mathbf{w}_i^l(t)]} = \frac{d[\varepsilon^2(t)]}{d[s_i^l(t)]} \cdot \frac{d[s_i^l(t)]}{d[\mathbf{w}_i^l(t)]} = \delta_i^l(t) \cdot \boldsymbol{\varphi}^l(\mathbf{x}^l(t)) \quad (l = 1, \ldots, L; i = 1, \ldots, N_{\mathrm{S}}^l), \quad (8)$$

ami a kívánt derivált. Figyeljük meg, hogy a derivált a $\delta_i^l(t)$ szám és az aktuális $\mathbf{x}(t)$ input l. rétegére eső $\mathbf{x}^l(t)$ lenyomatának $\boldsymbol{\varphi}^l(\mathbf{x}^l(t))$ reprezentációjával kifejezhető.

3.1.2. Regularizációs tag

Ez a tag egyszerűen megadható:

$$\frac{d[r(t)]}{d[\mathbf{w}_i^l(t)]} = \frac{d\left[\sum_{l=1}^{L}\sum_{i=1}^{N_S^l}\lambda_i^l \cdot \left\|\mathbf{w}_i^l(t)\right\|_{\mathcal{H}^l}^2\right]}{d[\mathbf{w}_i^l(t)]} = 2\lambda_i^l \cdot \mathbf{w}_i^l(t) \qquad (9)$$
$$(l = 1, \ldots, L; i = 1, \ldots, N_S^l).$$

Vegyük észre, hogy a derivált az aktuális $\mathbf{w}_i^l(t)$ súlyok skalárszorosa. Ezen forma szerint implicit hangolási szabály adható.

3.1.3. Költség tag

Használva a

$$\frac{d[c(t)]}{d[\mathbf{w}_i^l(t)]} = \frac{d[\varepsilon^2(t)]}{d[\mathbf{w}_i^l(t)]} + \frac{d[r(t)]}{d[\mathbf{w}_i^l(t)]} \quad (l = 1, \ldots, L; i = 1, \ldots, N_S^l)$$

összefüggést és az approximációs, illetve regularizációs tagokra kapott eredményeinket [(8) és (9) egyenlet] a

$$\mathbf{w}_i^l(t+1) = \mathbf{w}_i^l(t) - \mu_i^l(t) \cdot \frac{d[c(t)]}{d[\mathbf{w}_i^l(t)]} \quad (l = 1, \ldots, L; i = 1, \ldots, N_S^l)$$

legmeredekebb lejtő szabályban adódik, hogy

$$\begin{aligned}\mathbf{w}_i^l(t+1) &= \mathbf{w}_i^l(t) - \mu_i^l(t) \cdot \left(\delta_i^l(t) \cdot \boldsymbol{\varphi}^l(\mathbf{x}^l(t)) + 2\lambda_i^l \cdot \mathbf{w}_i^l(t)\right) \\ &= (1 - 2\mu_i^l(t) \cdot \lambda_i^l) \cdot \mathbf{w}_i^l(t) - \mu_i^l(t) \cdot \delta_i^l(t) \cdot \boldsymbol{\varphi}^l(\mathbf{x}^l(t)) \\ &\quad (l = 1, \ldots, L; i = 1, \ldots, N_S^l).\end{aligned}$$

Ugyanez duális formában

$$\boldsymbol{\alpha}_i^l(t+1) = \left[\left(1 - 2\mu_i^l(t) \cdot \lambda_i^l\right) \cdot \boldsymbol{\alpha}_i^l(t); -\mu_i^l(t) \cdot \delta_i^l(t)\right] \quad (l = 1, \ldots, L; i = 1, \ldots, N_S^l).$$

Így a hálózat súlyvektorainak kifejtési tuladjonsága [(3) egyenlet] az iterációk során öröklődik. Speciálisan, a számítás végeztével kapott $\mathbf{w}_i^l$ paraméterekre is fennáll. Összefoglalva, MLK-ra létezik visszaterjesztési eljárás. A levezetett explicit és implicit eljárásokat a 3.1. és a 3.2. táblázat foglalja össze.

4. Konklúziók

Új többrétegű modell, a Többrétegű Kerceptron (MLK) elméleti leírásával foglalkoztunk. Ez a hálózat egyesítheti a Többrétegű Perceptron (MLP) és a Támasztó Vektor Gépek (SVM) előnyeit: (i) Súlyai hangolhatók és a hangolás regularizációs

elvek mentén is megtehető. (ii) MLK-ban kernelek használata lehetséges. (iii) Az MLK hálózat behangolt súlyai segítségével a hálózat kimenete gyorsan számolható. (iv) Az MLK *rejtett rétegekkel* rendelkezik, és így képes lehet az SVM-ek adta partícionálásokat kombinálni. A megközelítés különböző adatbázisokon adódó előnyei és hátrányai jövőbeni kutatásaink tárgyát képezi.

Hivatkozások

[1] S. Haykin: *Neural Networks.* Prentice Hall, New Jersey, USA (1999)

[2] N. Aronszajn: *Theory of Reproducing Kernels.* Trans. of Am. Math. Soc. **68**, (1950) 337–404.

[3] V. N. Vapnik: *The Nature of Statistical Learning Theory.* Springer-Verlag New York, Inc., (1995)

[4] V. N. Vapnik: *Statistical Learning Theory.* Wiley, Chichester, GB, (1998)

[5] R. Herbrich: *Learning Kernel Classifiers.* MIT Press, (2002)

[6] K.-R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen and V. Vapnik: *Predicting Time Series with Support Vector Machines.* Advances in Kernel Methods, MIT Press, (1999), 243–254.

[7] G. Wahba: *Support Vector Machines, Reproducing Kernel Hilbert Spaces, and Randomized GACV.* Advances in Kernel Methods, MIT Press, (1999), 69–88.

[8] T. Evgeniou, M. Pontil and T. Poggio: *Regularization Networks and Support Vector Machines.* Advances in Computational Mathematics **13** 1, (2000), 1–50.

[9] V. Vapnik, S. Golowich and A.'Smola: *Support Vector Method for Function Estimation, Regression Estimation and Signal Processing.* Neural Information Processing Systems Vol. **9.**, MIT Press, Cambridge, MA, (1997)

[10] B. Schölkopf and A. J. Smola: *Learning with Kernels.* MIT Press, Cambridge, MA, (2002)

[11] J. Shawe-Taylor and N. Cristianini: *Kernel Methods for Pattern Analysis.* Cambridge University Press, (2004)

(Beérkezett: 2006. március 31.)

SZABÓ ZOLTÁN, LŐRINCZ ANDRÁS
EÖTVÖS LORÁND TUDOMÁNYEGYETEM
INFORMÁCIÓS RENDSZEREK TANSZÉK, INFORMATIKAI KAR
1117 BUDAPEST, PÁZMÁNY PÉTER SÉTÁNY 1/C.
szzoli@cs.elte.hu, andras.lorincz@elte.hu

MULTILAYER KERCEPTRON

Zoltán Szabó and András Lőrincz

Multilayer Perceptrons (MLP) are formulated within Support Vector Machine (SVM) framework by constructing multilayer networks of SVMs. The coupled approximation scheme can take advantage of generalization capabilities of the SVM and the combinatory feature of the hidden layer of MLP. The network, the Multilayer Kerceptron (MLK) assumes its own backpropagation procedure that we shall derive here. Tuning rule will be provided for quadratic cost function, with regularization capability as well. A further appealing property of our approach is that by the aid of the so called kernel trick the MLK computations can be performed in the dual space.