Лекция 15. Невронни мрежи от хиперсферични функции

15.1. Увод

Повечето от съществуващи невронни мрежи могат да бъдат класифицирани на два основни класа: 1) тези, които разделят входното пространство от примери на отворени области (обикновено – хиперплоскости) и 2) тези, които разделят това пространство на затворени области (хиперсфери, хиперпаралелепипеди и т.н.). Първият клас включва в себе си стандартни мрежи от персептроно-подобни възли, които обикновено са мрежи с право разпространение на активация, обучавани чрез алгоритми, в основа на които лежи алгоритъм Backpropagation (виж лекция 8). Такива мрежи имат голямо време за обучение породено от невъзможността техните тегла да бъдат инициализирани със стойности, лежащи близо до търсеното решение. От другата страна тяхното предимство е, че позволяват обучение чрез прилагане на техники за минимизация на грешка, така че при наличие на достатъчното количество обучаващи данни класификационната точност на изходите на такива мрежи приближава стойностите на Бейсови условни вероятности.

Вторият тип мрежи представляват хиперсферични класификатори (вероятностни невронни мрежи, мрежи на редуцираната Кулонова енергия (RCE) и др.). Такива мрежи не осигуряват минимизацията на грешката. Обаче, те се обучават много бързо, тъй като всеки техен скрит възел може да бъде инициализиран по такъв начин, че да представлява един прототип, заграждащ определена област на решение от входното пространство на примери. Този начин на изграждане на мрежата гарантира, че нейния изход се намира близо до доброто решение.

В настоящата лекция ще разгледаме основните видове на такива хиперсферични невронни мрежи.

15.2. Бейсова теория за класификация и сферични функции

Съгласно на Бейсовата теория за класификация (виж лекции 9-10), за получаване на оптималната класификация всеки нов пример $\boldsymbol{x_q} = (x_1,\ldots,x_p)$ трябва да бъде класифициран като принадлежащ към класа c_{MAP}:

$$c_{MAP} = \arg\max_{c_i \in C} P(c_i \mid \mathbf{x}_q)$$

където $C = \{c_1,\ldots, c_k\}$ – множество от възможни класове (дискретни стойности на целевия атрибут).

Прилагайки теоремата на Бейс, оптималната класификация на примера се задава от формула:

$$c_{MAP} = \arg\max_{c_i \in C} P(c_i \mid \mathbf{x}_q) = \arg\max_{c_i \in C} P(\mathbf{x}_q \mid c_i) P(c_i)$$

като и двете вероятности, участващи в тази формула, могат да бъдат приближено изчислени чрез използване на обучаващите данни.

Едно сравнително добро приближение към стойността на априорната вероятност $P(c_i)$ на класа c_i може лесно да бъде получено чрез изчисляване на честотата на срещане на обучаващите примери от съответния клас в цялото множество от обучаващите примери. Не е така стои въпрос с изчисляване на втората вероятност – вероятността на срещане на конкретния пример $\boldsymbol{x_q}$ сред всички възможни примери от този клас. С други думи, за всеки клас c_i ние трябва да намерим начин за апроксимиране на съответната функция на вероятностното разпределение на примери в този клас – $P(\mathbf{x}|c_i)$ или (предполагайки, че примерите се описват с непрекъснати атрибути) – функцията на плътността на вероятност $p(\mathbf{x}|c_i)$.

Разглеждайки тази задача като задача за апроксимация на неизвестна непрекъсната функция $f(\mathbf{x})$, едно от възможните решения на тази задача дава представянето на търсената функция като *линейна комбинация* от крайното множество фиксирани функции (наричани *базови* или *ядра*):

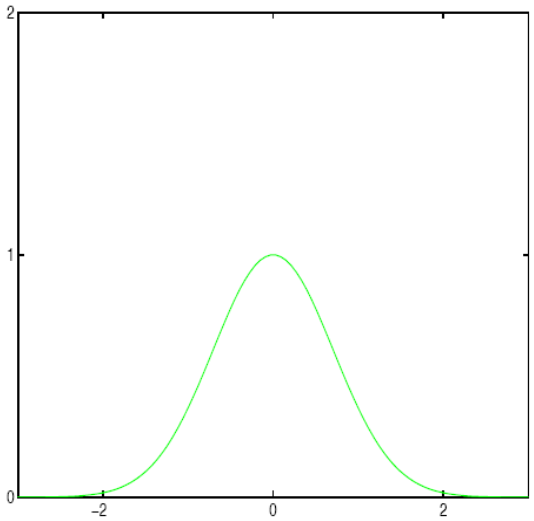
$$f(\mathbf{x}) = \sum_{j=1}^{m} w_j h_j(\mathbf{x})$$

Един специален клас на такива базови функции са така наречени *сферични функции (radial functions)*, стойността на които монотонно се намалява (или се увеличава) в зависимост от разстояние до централната точка на функцията. Една типична сферична функция е Гаусова функция, която в едномерен случай има вид:

$$h(x) = \exp(-\frac{(x-c)^2}{\sigma^2})$$

където c е централната точка на сферата, а σ - е нейния радиус.

Пример на такава функция с център $c = 0$ и радиус $\sigma = 1$ е показан на Фиг. 15-1.



Фиг. 15-1. Пример на сферична Гаусова функция

Американският статистик Емануил Парзен (Parzen 1962) е доказал, че функцията на плътност на вероятност на разпределение на примери в клас c_i може да бъде добре апроксимирана чрез линейната комбинация от сферичните функции, която в едномерен вариант се задава от формулата:

$$p(x \mid c_i) = \frac{1}{n_i \sigma} \sum_{j=1}^{n_i} h\left(\frac{x - x_j}{\sigma}\right)$$

където:

n_i – е броя на обучаващите примери от клас c_i

h – е избраната сферична функция

x_j - е обучаващ пример от клас c_i ($j = 1, \ldots, n_i$)

σ - е изглаждащ параметър (радиус на сферичната функция).

Резултатите на Парзен са били обобщени на случай на многомерни данни (Cacoullos (1966), като при използване на Гаусиани в качеството на сферичните функции формулата приема следния вид:

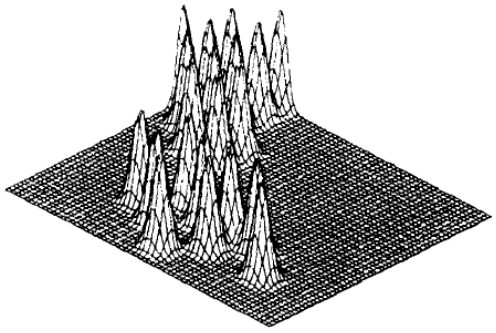
$$p(\mathbf{x} \mid c_i) = \frac{1}{(2\pi)^{p/2} n_i \sigma^p} \sum_{j=1}^{n_i} \exp\left(- \frac{\| \mathbf{x} - \mathbf{x}_j \|^2}{2\sigma^2} \right),$$

където $\| \mathbf{x} - \mathbf{x}_j \|^2 = \sum_{k=1}^{p} (x_k - x_{jk})^2$

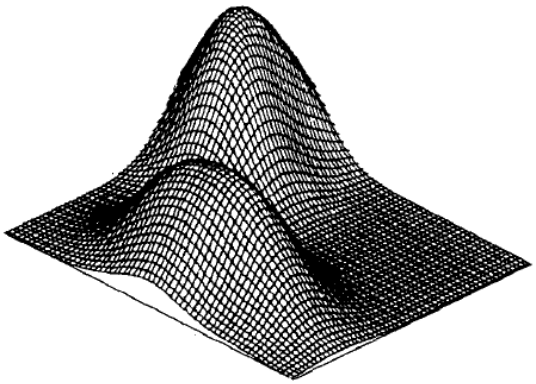
Тъй като при избор на оптималната класификация c_{MAP} константите в предишната формула са едни и същи за всички класове, тази формула може да бъде преписана във вида:

$$p(\mathbf{x} \mid c_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} \exp\left(- \frac{\| \mathbf{x} - \mathbf{x}_j \|^2}{2\sigma^2} \right) \quad (15.1)$$

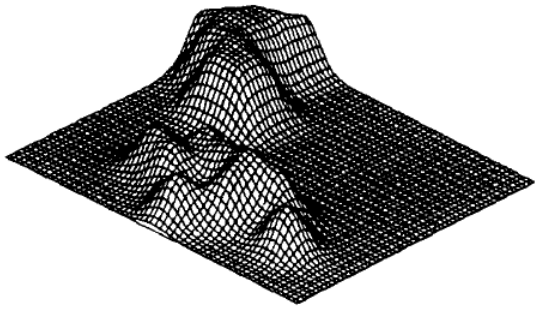
И така, плътността на вероятностното разпределение на примери в някой клас може да бъде представена като сума от Гаусиани, центрирани във всеки обучаващ пример от този клас. Тази функция може да апроксимира не само Гаусиана, а всяка гладка функция. На следващата фигура е показан ефект от различните стойности на изглаждащия параметър σ върху апроксимираната функция в случая, когато случайната величина ***x*** има само две размерности.



Малка стойност на σ



Още по-голяма стойност на σ



По-голяма стойност на σ

Фиг. 15-2. Влиянието на изглаждащия параметър σ върху вида на плътността на вероятност

Графиките на плътността са направени в съответствие с формула (15.1) за 3 различни стойности на σ, като във всички случаи се използват едни и същи обучаващи примери. Малката стойност на σ кара функцията, апроксимираща плътността на вероятност, да има множество различни моди, отговарящи на положението на обучаващите примери. По-голямата стойност води до по-голямата интерполация между точките. В този случай стойностите на ***x***, които са по-близки до обучаващите примери, са оценени като имащи почти същата вероятност на срещане като и самите обучаващите примери. Още по-голямата стойност на изглаждащия параметър води до още по-голямата степен на интерполиране. Една много голяма стойност на този параметър ще накара оценяваната функция на плътността да изглежда като Гаусиана не зависимо от истинския вид на функцията за вероятностното разпределение.

Връщайки се към задачата за оптималната класификация с представянето на плътността на вероятностното разпределение чрез сферичните функции (15.1) получаваме:

$$
\begin{aligned}
c_{MAP} &= \arg\max_{c_i \in C} P(c_i \mid \mathbf{x}_q) = \arg\max_{c_i \in C} P(\mathbf{x}_q \mid c_i) P(c_i) = \\
&= \arg\max_{c_i \in C} \frac{1}{n_{ci}} \sum_{j=1}^{n_{ci}} \exp\left(-\frac{\| \mathbf{x} - \mathbf{x}_j \|^2}{2\sigma^2} \right) \frac{n_{ci}}{n} = \\
&= \arg\max_{c_i \in C} \frac{1}{n} \sum_{j=1}^{n_{ci}} \exp\left(-\frac{\| \mathbf{x} - \mathbf{x}_j \|^2}{2\sigma^2} \right) = \arg\max_{c_i \in C} \sum_{j=1}^{n_{ci}} \exp\left(-\frac{\| \mathbf{x} - \mathbf{x}_j \|^2}{2\sigma^2} \right)
\end{aligned}
\qquad (15.2)
$$

15.3. Вероятности невронни мрежи

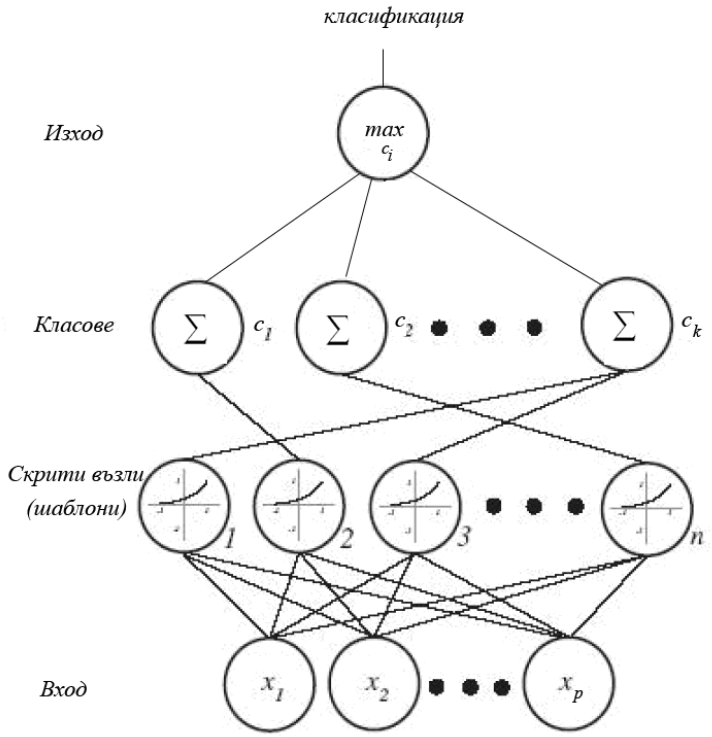
Вероятностните невронни мрежи (ВНМ) е мрежовата имплементация на вероятностния подход за класификация чрез апроксимация на функцията на плътност на вероятност чрез сферичните функции (15.2). Една ВНМ представлява многослойна невронна мрежа с право разпространение на активация, която се състои от четири слоя (Фиг. 15-2):

- *Входен слой*, който съдържа *p* неврона (*p* – броят на атрибути в обучаващи примери)
- *Скрит слой (шаблони)*, който съдържа *n* неврона (*n* е броя на обучаващите примери)
- *Сумиращ слой (класове),* който съдържа *k* неврона (*k* е броя на възможни класове)
- *Изходен слой*, който съдържа само един неврон, представляващ крайната класификация на входния пример.

Всеки скрит възел на ВНМ реализира избраната сферична функция, която в повечето случаи е Гаусиана, центрирана в един от обучаващите примери:

$$
g_{c_i}^{j}(\mathbf{x}) = \exp\left(-\frac{\| \mathbf{x} - \mathbf{x}_j \|^2}{2\sigma^2} \right), \quad j = 1,\ldots,n_{c_i}
$$

Всички обучаващи примери са представени като скритите възли на мрежата.



Фиг. 15-2. Архитектура на едно вероятностна невронна мрежа

Важната характеристика на ВНН е, че в нея се подразбира, че и обучаващите, и тестовите примери са *с единична дължина*, т.е. ||**x**|| = 1[1]. Това води до следното представяне на активиращата функция в скритите неврони:

$$g_{c_i}^{j}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_j\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\|\mathbf{x}\|^2 - 2\mathbf{x}\cdot\mathbf{x}_j + \|\mathbf{x}_j\|^2}{2\sigma^2}\right) = \exp\left(-\frac{2(1-\mathbf{x}\cdot\mathbf{x}_j)}{2\sigma^2}\right) = \exp\left(\frac{z-1}{\sigma^2}\right)$$

където z е скаларното произведение на нормираните вектори **x** и $\mathbf{x}_j$. Следователно, активиращата функция приема следните стойности:

$$g_{c_i}^{j}(z) = \begin{cases} 1, & \text{ако } z = 1 \\ e^{-\frac{1}{\sigma^2}}, & \text{ако } z = 0 \\ e^{-\frac{2}{\sigma^2}}, & \text{ако } z = -1 \end{cases}$$

Изчисляването на $z = \sum_{m=1}^{p} x_{jm} x_m$ става много бързо и лесно, като тегловния коефициент w_{jm} на връзка, свързваща *m*-я елемент x_m на входния вектор **x** = (x_1, ..., x_m,.. x_k) със скрития неврон с център в обучаващия пример $\boldsymbol{x}_j$ се установява равен на x_{jm}.

Всеки възел от сумиращия слой представлява един от възможни класове и е свързан със всички скрити възли, центрирани в обучаващите примери *от този клас*. След сумирането на всички активиращи сигнали на свързани с такъв възел скритите неврони,

[1] При подобна нормализация два паралелни вектора с различна дължина се преобразуват в един и същ вектор. За да бъде избегнато това, преди нормализацията всеки пример се разширява с един допълнителен *p+1* атрибут със стойността 1. След това този *p+1*-мерен вектор се нормализира, за да стане с дължина 1.

като изход се получава приближението на условната вероятност на съответния клас при зададения пример $p(c_i|\boldsymbol{x})$ чрез апроксимацията по Парзен на функцията на плътност на вероятност за съответния клас $p(\boldsymbol{x}|c_i)$.

Изходният възел осъществява крайната класификация на проверявания пример, прилагайки функция $\max_{c_i} p(c_i \mid \mathbf{x})$.

15.3.1. Обучение на ВНМ

Обучението на една ВНН се състои от две части – 1) избор на архитектурата на мрежата и 2) избор на параметъра на σ. Архитектурата на мрежата се определя напълно от обучаващото множество и броя на класове. Всеки от възможните класове съответства на един сумиращ възел, който се свързва със всички скрити възли, представящи обучаващите примери от този клас. Всеки скрит възел се свързва със всички входни възли, като теглата на връзки w_{ij} се установяват равни на нормираните стойности на съответните атрибути на асоциирания с възела обучаващ пример $\mathbf{x}_i$.

Алгоритъм 1 (обучение на ВНМ)
Вход: $D = \{<\boldsymbol{x}_1,c_1>, \ldots, <\boldsymbol{x}_n, c_n>\}$
begin
инициализация: $j = 0$, n = брой обучаващи примери
do $j := j + 1$
нормализация: $x_{jk} = \dfrac{x_{jk}}{\sqrt{\sum_{i=1}^{p+1} x_{ji}^2}}$
обучение: $w_{jk} := x_{jk}$, $k = 1, \ldots, p+1$
$a_{jc} := 1$ (връзка на скрит възел $\mathbf{x}_j$ със сумиращ възел c)
until $j = n$
end

Алгоритъм 2 (Класификация с ВНМ)
begin
инициализация: $k = 0$, $g_c = 0$ за всички класове c; $\mathbf{x}$ – нормализирания тестов пример
do $k := k + 1$
$z_k = \mathbf{x} \cdot \mathbf{w}_k$
if $a_{kc} = 1$ **then** $g_c = g_c + e^{\frac{z_k - 1}{\sigma^2}}$
until $k = n$
върни $class = \arg\max_c \{g_c\}$
end

Важно е да се помни, че класификационното поведение на една ВНН зависи от качеството на избраното обучаващо множество. То трябва да бъде една действително представителна извадка от генералната съвкупност на примери от дадената предметна област. От тази гледна точка изискванията на една ВНН към качеството на примери, използвани за нейното обучение, са доста по-големи от тези, които предявява една

невронна мрежа, обучавана с алгоритъма за обратното разпространение на грешката (backpropagation). Тъй като крайното решение в една ВНН се приема на базата на приноси на всички обучаващи примери, такава мрежа е толерантна към наличието на шума и крайности в обучаващото множество, както и към разпръснатостта на самите обучаващи примери.

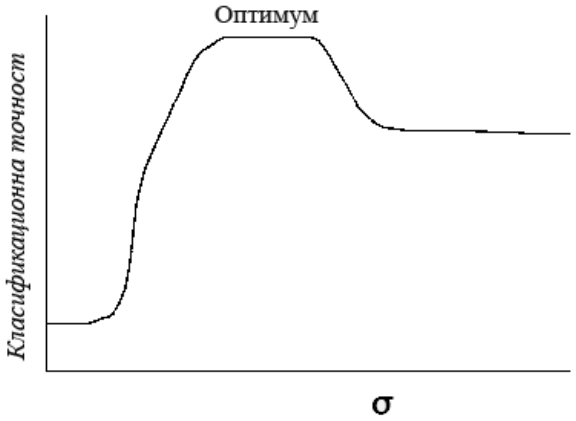
Важната характеристика на ВНМ е, че те, на практика, почти не изискват повторното обучение при добавяне на нови обучаващи примери или при премахване на част от тях, тъй като в тези случаи просто се добавят или се премахват скритите възли в мрежата.

При нарастване на размера на обучаващото множество класификационната точност на ВНН асимптотично се приближава към точността на оптималния Бейсов класификатор, а апроксимираната чрез мрежата функция на плътността на вероятност се приближава до истинската плътност на вероятност при предположение, че тази плътност представлява една гладка функция.

На практика, процесът на обучение на една ВНМ се свежда до определяне на стойността на изглаждащия параметър σ. Този процес е доста бърз – на няколко порядъка е по-бърз от обучение на една невронна мрежа чрез алгоритъма backpropagation.

В общия случай начините за определяне на оптималната стойност на σ са два – обоснован избор, базиращ се на предварителните знания за проверяваната проблемна област, или намиране на някоя приближена стойност чрез прилагане на евристични техники.

Един от такива евристични подходи се състои в прилагането на техниката за крос-валидацията „един вън" (Leave-one–out (LOU) cross-validation) (Фиг. 15-3). Избира се определен интервал на възможните стойности на σ и оптималната стойност на този параметър се търси чрез по-стъпково стесняване на интервала. За всяка проверявана стойност се изчислява класификационната точност на мрежата чрез прилагането на LOU крос-валидацията: един от обучаващите примери се премахва и се прави опит той да бъде класифициран с помощта на останалите обучаващи примери. Този цикъл се повтаря със всеки пример от обучаващото множество, като на края се изчислява общия брой на правилно класифицирани примери.



Фиг. 15-3. Евристичен избор на стойността на σ

И така, предимствата на ВНМ са следните:

- Бърз процес на обучение – на практика необходимото време за обучение е на порядък по-малко от това, необходимо за обучение на „класическата" невронна мрежа.
- Позволява лесна паралелна имплементация.

- Гарантира сходимостта към оптималното решение при нарастване на размера на обучаващото множество.

Като недостатъци на ВНМ могат да бъдат отбелязани следните нейни свойства:

- Не е толкова обща като многослойна невронна мрежа, обучавана чрез алгоритъма backpropagation от гледната точка на създаваните от мрежата повърхнини на решения.
- Има големи изисквания към паметта.
- Сравнително бавно време, необходимо за класификация.
- Изисква наличие на представителното обучаващо множество.

Вероятностната невронна мрежа може да се разглежда като една мрежова имплементация на глобалния вариант на алгоритъма на k най-близки съседи, претеглени по разстояние, където коефициент на влияние на всеки съсед се определя както от неговото разстояние до проверявания пример, така и от един глобален изглаждащия параметър - σ.

15.4. Обобщени вероятностни мрежи

Вероятностните невронни мрежи представят областите от пространството, съответстващи на всеки клас, като съвкупност от хиперсфери с центрове във всеки обучаващ пример от дадения клас, като радиусът на всички сфери е един и същ. Едно естествено обобщение на този подход е да позволим броят на хиперсфери да не бъде фиксиран и всяка от хиперсферите да има различен радиус. Съществуващите подходи към тази задача могат да бъдат разделени на две групи – тези, в които центрове на хиперсферите съвпадат с някои от обучаващите примери, и тези, в които центрове на хиперсфери не съвпадат с никой от обучаващите примери.

15.4.1. Мрежи на ограничена Кулонова енергия

Мрежите на ограничена Кулонова енергия (Reduced Coulomb Energy (RCE) networks) са направени по аналогия с модела за движение на заредена положителна частица в тримерното пространство, съдържащо няколко отрицателно заредени частици (Reilly et al. 1982). Съществуват различни версии на алгоритъма за научаване на RCE мрежи. В общия случай един алгоритъм за създаване (научаване) на RCE мрежата е итеративен и се състои от следните стъпки – избор на центъра на хиперсфера (така наречения „прототип") за всеки клас, създаване на хиперсфера с фиксиран радиус (параметър на алгоритъма) около прототипа и след това свиване на сферите (ако е необходимо) по такъв начин, че всяка от тях да не съдържа нито един обучаващ пример от клас, отличен от този на съответния прототип. Като резултат пространството на примери се разбива на множество хиперсфери, съдържащи обучаващите примери само от един и същ клас.

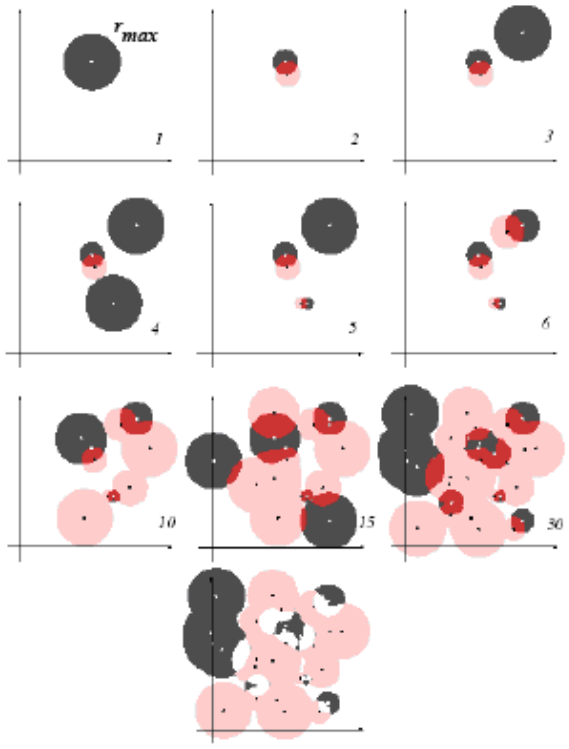
Във варианта на RCE алгоритъма, предложен от Duda и Hart (2001), прототипите (скритите възли на мрежата) съвпадат с обучаващите примери, а единствените параметри, които трябва да бъдат научени – са радиуси на хиперсферите (r_i). Примерът на RCE мрежа е показан на Фиг. 15-4

Алгоритъм за научаване на RCE мрежа (Duda and Hart 2001)

Вход: Множество от обучаващи примери D = {<$\mathbf{x}_1$,c_1>,…, <$\mathbf{x}_n$, c_n>}
Параметър r_{max} – максимално допустим радиус на една хиперсфера
ε - параметър с малка стойност
d(. , .) – Евклидово разстояние в пространството на примери

begin
инициализация: $k = 0$, n – брой на обучаващи примери, r_{max} – максимален радиус на хиперсфери, ε - параметър с малка стойност
do $k := k + 1$
научаване на тегла: $\mathbf{w}_{kj} := \mathbf{x}_{kj}$ ($j = 1,...,p$)
намери обучаващ пример $\mathbf{x}$, най-близък до $\mathbf{x}_k$, който не от същия клас c_k като на $\mathbf{x}_k$. $\mathbf{x} = \arg\min_{c \neq c_k}\{d(\mathbf{x}_k, \mathbf{x}')\}$
избор на радиус: $r_k = \min\{d(\mathbf{x}_k, \mathbf{x}) - \varepsilon, r_{\max}\}$
$a_{kj} := 1$ (връзка на скрит възел $\mathbf{x}_k$ със сумиращ възел c_k)
until $j = n$
end

Илюстрацията на алгоритъма за научаване на RCE мрежи за случая на двумерни точки е показан на Фиг. 15-4.



Фиг. 15-4. Със сиво са означени хиперсфери от първия клас, а с розово – от втория. С червено са означени „проблемни" области от пространството, покрити от хиперсфери, принадлежащи към различни категории. Числото вдясно показва броя на научените хиперсфери.

RCE мрежи класифицират неизвестни примери като принадлежащи към класа на хиперсфера, в който примерът попада. Ако примерът попада извън всички създадени

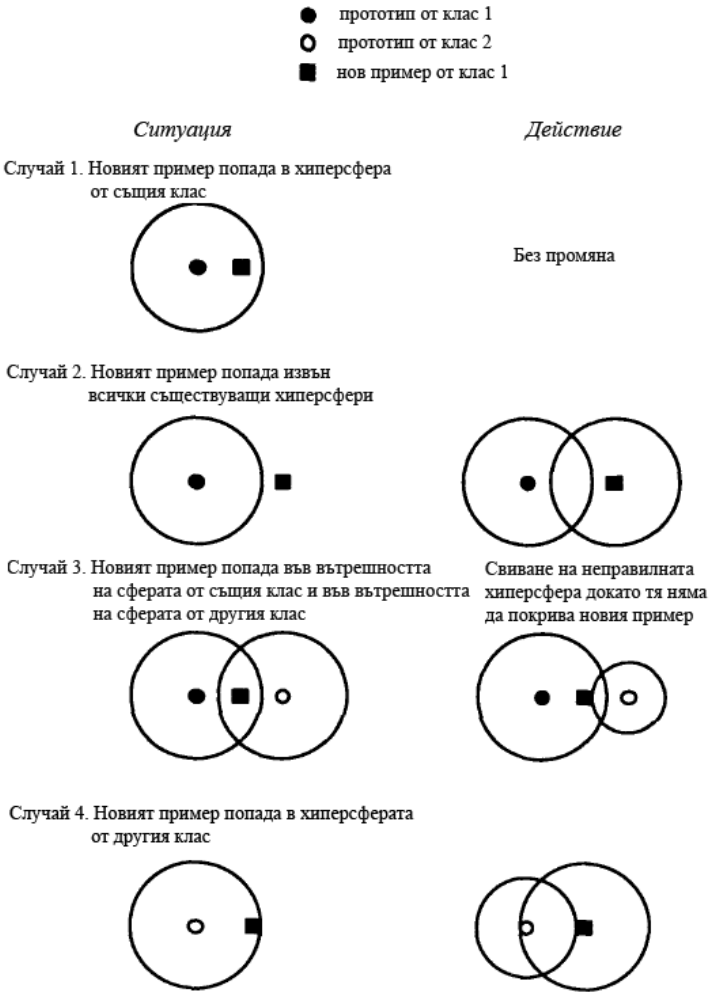
хиперсфери, то той се класифицира като „неизвестен”. Същото важи и за областите, в които се препокриват хиперсфери, принадлежащи към различни класове.

Алгоритъм за классификация с RCE мрежа (Duda and Hart 2001)

begin
инициализация: $k = 0$, **x** = тестовия пример, $S = \varnothing$
do $k := k+1$
if $d(\mathbf{x}, \mathbf{x}_k) < r_k$ **then** $S = S \cup c_k$
until $k = n$
if всички елементи (класове) в S са еднакви, то върни $c = class(S)$
else върни клас "*неизвестен*"
end

15.4.2. По-сложи варианти на RCE мрежи

Прототипите в RCE мрежи не е задължително съвпадат (както по мястото в пространство, така и по брой) с обучаващите примери. В този раздел ще разгледаме алгоритъм, в който в качеството на прототипи се избират *някои* от обучаващите примери. Идеята на създаване на хиперсфери е същата като и в предишния алгоритъм – всяка от хиперсферите трябва да покрива обучаващите примери само от един и същ клас. Научаването на RCE мрежата става на *епохи*, като всяка епоха се състои в итеративното представяне на всички обучаващи примери. За всеки пример се намират всички хиперсфери, в които той попада, и се проверява дали класът на примера съвпада с класа на прототипа, който се намира в центъра на съответната хиперсфера. Ако класовете съвпадат – не се прави нищо и се проверява следващия обучаващия пример. Ако класовете са различни, радиусът на сферата, която „неправилно” покрива проверявания



Фиг. 15-5. Илюстрация на основните стъпки на RCE алгоритъм

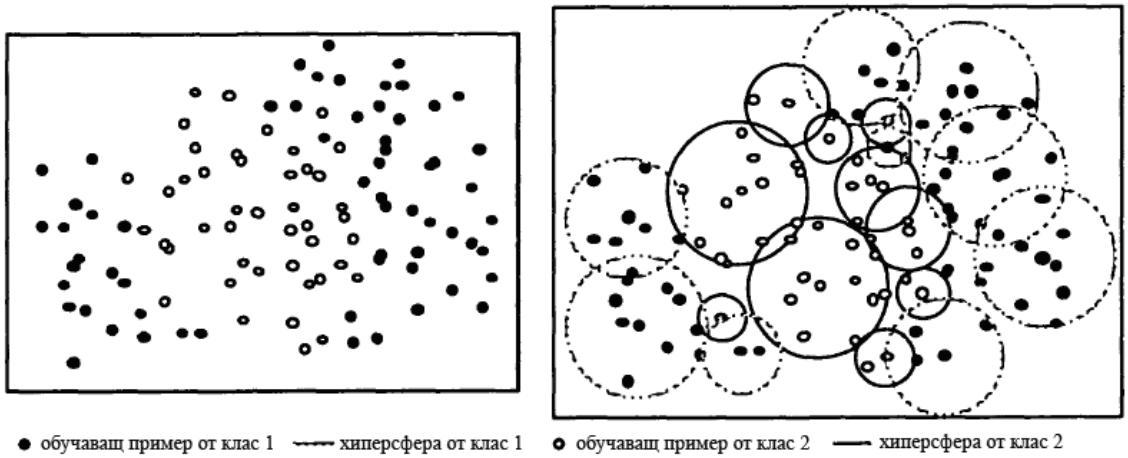
пример се намалява по такъв начин, че примерът остава да лежи върху границата на сферата. Ако примерът попада в област на пространството, която не е покрита от никакви хиперсфери, то примерът се разглежда като нов прототип и около него се създава хиперсфера с максимално допустим радиус. Работата на алгоритъма приключва, когато няма промени в радиусите на научените хиперсфери. Фиг. 15-5 илюстрира основните стъпки на алгоритъма.

Алгоритъм за научаване на RCE мрежа (Wilensky et al. 1993)

Вход: Множество от обучаващи примери D = {<$\mathbf{x}_1$,c_1>,…, <$\mathbf{x}_n$, c_n>}
Параметър r_{max} – максимално допустим радиус на една хиперсфера
d(. , .) – избраната функция на разстояние в пространство на примери
Чрез *S(**p**,c,r)* означаваме хиперсфера с радиус *r* и център в прототип ***p***, който принадлежи към клас *c*, а чрез *S* множество от всички хиперсфери.

begin
инициализация: $S = \varnothing$
repeat
for $\forall < \mathbf{x}, c > \in D$ **do**
for $\forall S(\mathbf{p}_i, c_i, r_i) \in S$ **do**
if $\mathbf{x} \in S(\mathbf{p}_i, c_i, r_i)$ and $c \neq c_i$ **then**
$r_i = d(\mathbf{p}_i, \mathbf{x})$
end if
end for
if $\forall S(\mathbf{p}_i, c_i, r_i) \in S \quad \mathbf{x} \notin S(\mathbf{p}_i, c_i, r_i)$ **then**
създай нова хиперсфера *S(**x**, c, r_{max})* **and** $S = S \cup S(\mathbf{x}, c, r_{\max})$
end if
end for
until няма промяна в *S*

Фиг. 15-6 илюстрира работата на алгоритъма за случая на двумерни точки, принадлежащи към два класа.



Фиг. 15-6. Илюстрация на RCE алгоритъма за случая на двумерни точки, принадлежащи към 2 класа

И при този вариант на алгоритъма създаваните RCE мрежи запазват своите основни характеристики:

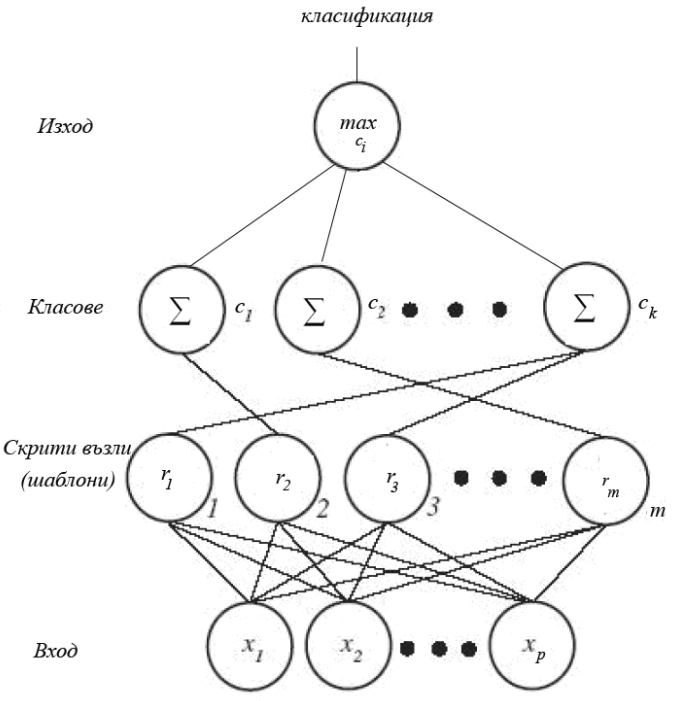
- Създадените хиперсферите могат да се препокриват.
- Не цялото пространство от възможни примери се покрива от хиперсферите.

Тези характеристики определят и начина на използването на RCE мрежата при класификация на новия пример:

- Когато примерът попада в областта на пространството, покрито от хиперсферите, принадлежащи само към един клас – той се класифицира като принадлежащ към същия клас.
- Когато примерът попада в зоната, която не е покрита от нито една хиперсфера, примерът се класифицира като принадлежащ към неизвестен (т.е. не съвпадащ с нито един от известни класове) клас.
- Когато примерът попада в зоната на конфликт, т.е. е покрит от няколко хиперсфери, принадлежащи към различни класове, неговата класификация остава неизвестна. Съществуват и различни варианти за решаване на конфликти – например Wang et al. (2006) предлагат в такива случаи да бъде избран клас на хиперсферата, за която отношението на разстояние на проверявания пример до центъра на сферата към радиуса на сферата е най-малко (т.е. примерът се намира относително най-дълбоко в тази сфера).

15.4.3. Реализацията на RCE алгоритъма във вид на невронна мрежа

При имплементацията на описаните по-горе алгоритми във вид на невронна мрежа трябва да отчитаме, че както тестовите примери, така и всички прототипи са представени във вид на вектори с единична дължина. При това положение Евклидовото разстояние между тестовия вектор и прототип е пропорционално на скаларното произведение на тези вектори. Със всеки прототип $\mathbf{p_i}$ можем да свържем параметър ρ_i, който съответства на радиуса на хиперсфера r_i в описаните по-горе алгоритми. Както и в случая на ВНМ теглата $\mathbf{w_i}$ на връзки, свързващи всеки скрит възел (съответстващ на прототип $\mathbf{p_i}$), с входните възли, се установяват равни на съответните атрибутни стойности на прототипа: $\mathbf{w_i} = \mathbf{p_i}$. При това положение при постъпване на входа на мрежата на някой тестов пример $\mathbf{x}_j$, входът към скритият възел i се определя като $Input_i = \rho_i \mathbf{w}_i \cdot \mathbf{x}_j$. Скритият възел i се свързва в мрежата с възела от третия слой, съответстващ на класа на съответния прототип $\mathbf{p_i}$, като теглото на тази връзка е равно на 1.



Фиг. 15-7. Пример на RCE мрежа

За да имитираме механизъм, при който скритият неврон се активира само ако тестовия пример попада в описаната около прототипа хиперсфера с радиус r_i, ще въведем един

праг w_0 и ще дефинираме следната активираща функция на скрития неврон, определяща неговия изход:

$$Output_i = \begin{cases} \rho_i \mathbf{p}_i \cdot \mathbf{x}_j, \text{ако } \rho_i \mathbf{p}_i \cdot \mathbf{x}_j > w_0 \\ 0, \text{ако } \rho_i \mathbf{p}_i \cdot \mathbf{x}_j \le w_0 \end{cases}$$

Построяването на RCE мрежа става итеративно по следния алгоритъм:

Вход: Множество от обучаващи примери D = {<$\mathbf{x}_1$,c_1>,…, <$\mathbf{x}_n$, c_n>}
ρ_0 – началната стойност на параметра, аналогичен на r_{max}.
w_0 – константа, играеща роля на прага
S - множество от скрити възли в мрежата

инициализация: Избери по случаен начин пример <**x**,c>∈D и създай първия скрит неврон p_1, свързан със всички входни възли. Установи теглата на съответните връзки на: $\mathbf{w}_1 := \rho_0 \mathbf{x}$;
Създай на изходния слой *k* неврона, отговарящи на класове в D.
Свържи първия скрит неврон с изходния неврон, отговарящ на класа *c*, като установи теглото на тази връзка да е равно на 1.
D := D\{<x,c>}

repeat
Избери по случаен начин пример <$\mathbf{x}_i$,c_i>∈D. Пусни $\mathbf{x}_i$ като вход на създадената до момента мрежа.
if изходът на изходния неврон, съответстващ на класа c_i, е равен на нула
then
създай нов неврон $\mathbf{p}_i$ в скрития слой, свързан със всички входни възли.
Установи теглата на съответните връзки на: $\mathbf{w}_i := \rho_0 \mathbf{x}_i$;
Свържи новия скрит неврон със изходния неврон, отговарящ на класа c_i, като установи теглото на тази връзка да е равно на 1.
else
repeat
За всеки изходен възел с ≠ c_i, чийто изход е > 0 **do**
Намери скрит възел p_k, свързан изходния възел
Промени стойността на ρ_k по следния начин:

$$\rho_k := \frac{w_0}{\mathbf{p}_k \cdot \mathbf{x}_i}$$

endelse
D := D\{<x_i,c_i>}
until D = ∅

15.4.4. Алгоритъм за научаване на RCE мрежи на Wang et al. (2006)

RCE мрежи имат определени предимства в сравнение с класическите многонивови невронни мрежи (МНМ). Първо, тъй като алгоритмите за научаване на RCE мрежи са инкрементални, те не изискват предварителното знание за броя на скрити възли. Второ, алгоритмът за обучение на такива мрежи е лесен за имплементация и има гарантирана сходимост. Обикновено RCE мрежи изискват много малък брой минавания през обучаващите данни (епохи) преди да се получи сходимост към окончателното решение,

а получената мрежа е в състояние да решава задачи, в които класове са линейно несепарабелни. RCE мрежи успешно използвани за решаване на такива задачи като оценка на риска, разпознаване на ръкописен текст, класификация на пръстови отпечатъци и др. Обаче, съществуват ред тънкости, които трябва да бъдат взети под внимание при използването на RCE мрежи. Първо, поведението на мрежата зависи от параметъра r_{max}, определящ максималния размер на създаваните хиперсфери. Изборът на този параметър засяга броя на създаваните прототипи, времето за обучение и, което е най-важно, класификационната точност на мрежата. На практика стойността на този параметър се определя чрез крос-валидацията, която е доста скъп от изчислителната гледна точка метод. Второ, тъй като алгоритъм за обучение има стохастичен характер, точния вид на мрежата зависи от реда, в който се подават обучаващите данни. Освен това, хиперсферите могат да бъдат създадени върху границите на класове, което води до потенциални грешки при класификацията.

В настоящия раздел ще разгледаме един метод, който избягва указаните недостатъци на описаните по-рано алгоритми за научаване на RCE мрежи. Вместо последователното използване на обучаващите примери, този алгоритъм отначало използва геометричната информация за обучаващите данни с цел създаване на едно множество от потенциални кандидати-хиперсфери, а след това избира от тях едно подмножество от прототипи на базата на критерия за минимизиране на локалната вероятност на грешката или минимизиране на общия брой на създаваните хиперсфери. Разработените от автори алгоритми са лесни за имплементиране, не се нуждаят от параметри за нагласяване и, по данните на авторите, постигат по-добри резултати в сравнение с „класически" алгоритми за научаване на RCE мрежи както върху изкуствени, така и върху реални бази данни.

Първият вариант на алгоритъма опитва да минимизира вероятността за грешна класификация чрез увеличаване на убедеността в класификацията, която се предлага от една хиперсфера, а тази убеденост е в пряка зависимост от броя на обучаващите примери от един и същ клас, покрити от хиперсферата.

Алгоритъм 1 за научаване на RCE мрежа (*Wang et al. 2006*)

Вход: Множество от обучаващи примери D = $\{<\mathbf{x}_1,c_1>,\ldots,<\mathbf{x}_n,c_n>\}$
d(. , .) – избраната функция на разстояние в пространство на примери
Чрез $S(\mathbf{p},c,r)$ ще означаваме хиперсфера с радиус r и център в прототип $\mathbf{p}$, който принадлежи към клас c, а чрез S множество от всички хиперсфери.

begin
инициализация: $S=\varnothing$
for $\forall <\mathbf{x},c>\in D$ **do**
създай $S(\mathbf{x},c,r)$: $r=\min_{c_i\neq c} d(\mathbf{x}_k,\mathbf{x})$
изчисли броя на обучаващи примери, попадащи в $S(\mathbf{x},c,r)$
end for
for $\forall <\mathbf{x},c>\in D$ **do**
намери сфера $S(\mathbf{p},c,r)$: $\mathbf{x}\in S(\mathbf{p},c,r_i)$, която съдържа най-голям брой примери от клас c
if $S(\mathbf{p},c,r)\notin S, S=S\cup S(\mathbf{p},c,r)$
end if
end for

Алгоритмът се състои от два цикла – по време на първия покрай всеки обучаващ пример се създава хиперсфера (потенциален кандидат за крайното решение) с радиус, равен на разстояние до най-близък до центъра на сферата обучаващ пример от друг (отличен от този на центъра) клас. За всяка създадена хиперсфера се изчислява броят на покритите от нея примери. Тъй като убедеността в правилната класификация на тестов пример, попаднал в околността на сферата, е равен на корен квадратен от броя на обучаващите примери, покрити от сферата, колкото повече примери са покрити от хиперсферата, толкова по-малка е вероятността на нейната класификационна грешка. По тази причина при втория цикъл от алгоритъма от множеството на потенциални кандидати-хиперсфери се избират последователно тези, които съдържат най-много обучаващи примери. Полученото множество от хиперсфери покрива „най-убедително" всички обучаващи примери, като броят на хиперсфери зависи само от взаимното разположение на обучаващите примери. Авторите нарекли този алгоритъм „алгоритъм за убедителното сферично покриване" Confident Sphere-Covering (CSC).

Вторият алгоритъм опитва се да минимизира броя на хиперсферите. Тъй като общата задача за намирането на минималното покритие чрез хиперсфери е NP-пълна, авторите предлагат следния евристичен подход към тази задачи:

Алгоритъм 2 за научаване на RCE мрежа (*Wang et al. 2006*)

Вход: Множество от обучаващи примери D = {<$\mathbf{x}_1$,c_1>,…, <$\mathbf{x}_n$, c_n>}
d(. , .) – избраната функция на разстояние в пространство на примери
Чрез *S(**p**,c,r)* ще означаваме хиперсфера с радиус *r* и център в прототип ***p***, който принадлежи към клас *c*, а чрез *S* множество от всички хиперсфери.

begin
инициализация: $S = \varnothing$
for $\forall < \mathbf{x}, c > \in D$ **do**
създай $S(\mathbf{x}, c, r)$: $r = \min_{c_i \neq c} d(\mathbf{x}_k, \mathbf{x})$
изчисли броя на обучаващи примери, попадащи в $S(\mathbf{x}, c, r)$
end for
повтори
намери сфера $S(\mathbf{p}, c, r)$**,** която съдържа най-голям брой *останалите непокрити* от *S* примери и я добави към *S*
until всички обучаващите примери са покрити

Първия цикъл е същия както и в Алгоритъм 1. На втория цикъл се избира хиперсферата, която покрива най-много от останалите примери, които още не покрити от избраните по-рано хиперсфери. Очевидно е, че в началото на тази стъпка се избира хиперсферата, която покрива най-много обучаващите примери. Този вариант на алгоритъма авторите нарекли „лаком алгоритъм за сферично покриване" - greedy sphere-covering (GSC).

За да избегнат отговор „неизвестна класификация" в случаи, когато тестовия пример попада в областта, покрита от хиперсфери с различна класификация, или в областта извън всички хиперсфери, автори предлагат използването на стратегията „победителят получава всичко":

Алгоритъм за класификация:

Вход:
x – тестовия пример
$S = \{S(\mathbf{p}_1,c_1,r_1), \ldots, S(\mathbf{p}_k,c_k,r_k)\}$ – набор от хиперсфери, покриващи всички обучаващи примери

Изход: $c = c_i,\ i = \arg\min_{j=1..,,k} \frac{d(\mathbf{x}, \mathbf{p}_j)}{r_j}$

Избира се класът на хиперсферата с най-малкото отношение разстояние до центъра – към радиус на сферата.

Експерименти върху 5 от бази данни от „класическото" хранилище на тестови данни (UCI repository) показали, че и двата алгоритъма са по-точни от „класическия" RCE алгоритъм (с най-добре настроен параметър r_{max}), както и от алгоритъма на най-близкия съсед. Най-добрата точност се постига от алгоритъма CSC, като броят на избраните хиперсфери е значително по-малък от общия брой на обучаващите примери и при 3-те варианта на RCE алгоритъма.