Teма 5.Изкуствени невронни мрежи

Изкуствените невронни мрежи са програмни или апаратни средства, симулиращи отделни функции на човешкия мозък.

**1.Биологичният феномен „човешки мозък”.**

Изследванията в областта на **неврологията** водят до забележителни открития. Установява се, че съвместното функциониране на голям брой неврони може да доведе до мислене, действия и съзнание или с други думи, мозъкът да породи разум. В сравнение с цифровия компютър[[1]](#footnote-1) човешкият мозьк работи около 100 000 пъти по- бързо и може да проявява творчество, въображение, интуиция и др., непостижими за съвременните машини. Мозъкът е така организиран, че невроните му изпълняват редица функции (като разпознаване на образи, перцепция на звук и др.) много пъти по- бързо от всеки цифров компютър днес. Човек разпознава лицата на близките си за около 0.1 секунда. При тази скорост системата от неврони, която прави разпознаването, извършва най- много 100 стъпки на преминаване от поляризирано в деполяризирано състояние и обратно. За 100 стъпки един цифров процесор не може да пресметне почти нищо съществено.

Предполага се, че изключителните възможности на човешкия мозък се дължат на огромния брой неврони в него, около 1000 пъти повече от логическите елементи в компютъра, на начина на свързването им, както и на паралелната им работа.

Човешкият мозък може да възприема 4 милиарда бита информация в секунда, докато при вербалните взаимоотношения ние приемаме само 500 бита информация за секунда. (Eдна славянска буква е приблизително 5 бита.)

При своята работа мозъкът е много ефективен и по отношение на енергията. Той изразходва около 10-16 джаула за операция на секунда. За сравнение, компютърът изразходва 10-6 джаула за операция на секунда. Мозъкът има способността да изгражда свои собствени правила чрез опитност. Опитността се натрупва през годините, като най- драматичното му развитие е през първите 2 години след раждането. През този ранен период на развитие се формират около 1 милион връзки (синапси) на секунда.

Според неврофизиологията човешкият мозък се състои от приблизително 1011 неврона (**Фигура** 1), които са свързани с около 1014 нервни връзки.

Дендрити



вход

Синапс

изход

вход

Аксон

Фигура 1. Схематично изображение на биологичен неврон

Всеки неврон е изграден от тяло, аксон и дендрити. Тялото изпълнява биохимични трансформации, необходими за съществуването на неврона. Аксонът е линията за предаване, представляващ клетъчно влакно с гладка повърхност, малко разклонения и голяма дължина. Той носи изходните сигнали от клетката. Аксонът е покрит с клетки от органично вещество миелин, които изолира предаването на електрически сигнали по влакното.

Дендритите са къси фини разклонения, през които невроните приемат сигнали. Понеже тези сигнали са аналогови, напрежението варира. Ако то е в определени граници, невронът се възбужда и предава нов аналогов сигнал.

**Синапсите са елементарни структурни и функционални единици, които предават импулси между невроните.** Човешкият мозък има около 60 трилиона синапси. В традиционните описания на невронната организация е прието, че синапсът е проста връзка, която може да наложи възбуждане или подтискане на възприемащия неврон.

Невронът „взема решение” чрез възбуждане или неактивиране. Тези решения са на много ниско ниво. Необходимо е да бъдат взети голям брой решения от много неврони дори само, за да се прочете едно изречение. Решенията на по- високо ниво са резултат от колективните входове и изходи на много неврони. За адаптиране към околната среда се създават нови синапси между невроните и се изменят съществуващите.

Известно е, че общият брой неврони не се променя съществено в течение на човешкия живот. Почти еднакъв е броят на невроните в мозъка на учения, политика, спортиста. Разликата е в силата и електрическата проводимост на нервните влакна, съединяващи невроните. На тази основа е хипотезата, че всички знания, мисли, емоции в човешкия мозък са кодирани във вид на синаптични връзки. Процесът на обучение на човека в продължение на целия му живот се състои в коригиране съдържанието на матрицата от тези връзки.

Невроните получават входни сигнали от външното обкръжение или от други неврони. Невроните са свързани помежду си чрез дендритите в сложна мрежа, по която се предават сигнали. Приетите от синапсите входни сигнали се сумират. Това е сложен физико- химичен процес, при който се повишава или намалява електрическият потенциал на неврона. Ако той премине определено прагово ниво, се изпраща сигнал с определена продължителност и сила и тогава невронът се счита за възбуден. В това състояние той изпраща сигнал към други неврони. Друга част от сигналите има за цел да възпрепятства този процес. След известен период на покой, невронът може да получи ново активиране.

Характерно за функционирането на нервната система е, че тя може да се обучава и адаптира и всички неврони действат паралелно, което обяснява високата скорост на реагиране при човека.

Към феноменалните способности на мозъка се отнасят :

* Представянето на информацията за външния свят;
* Краткосрочно и дългосрочно съхранение и извличане на информация;
* Асоциативно търсене и самоорганизация на паметта;
* Мигновеното разпознаване на лица;
* Творческото озарение.

Все още няма конструктивно научно обяснение на тези способности.

**2.** **Същност на изкуствените невронни мрежи**.

За да се моделират някои от функциите на нервната система се правят редица опростявания и/или въвеждат ограничения. Например, не се постига забавянето на сигнала в биологичните системи, което е в резултат на функцията на задържане. Също така има известни процеси на синхронизация, които се считат за много важни, но не са моделирани по изкуствен път. Само биологичните неврони могат да реагират на непрекъснати входни сигнали. Въпреки ограниченията, изкуствените невронни мрежи приличат по конструкцията и работата си на естествените. Чрез свързването на невроните и чрез тегловни коефициенти на връзките между тях, се изграждат изкуствени невронни мрежи.

**Изкуствените невронни мрежи са динамични системи, изградени от свързани помежду си във вид на насочен граф елементарни процесори, наречени изкуствени неврони и способни да генерират и предават изходна информация в отговор на входно въздействие.**

Изкуственият неврон (възел) може да се разглежда като модел, на биологичния неврон. Състои се от четири основни елемента:

* + множество синаптични връзки, всяка от които се характеризира с тегло (синаптично тегло);
  + суматор на входните сигнали;
  + активираща функция за ограничаване на амплитудата на сигнала на изхода от неврона. Амплитудата е в затворения интервал [0,1] или [-1,1].
  + oтклонение (bias), коeто има за задача поддържане на входно ниво на активиращата функция, както и увеличаване или намаляване на сигнала към нея. Този елемент не е задължителен за всички мрежи.

В литературата и практиката вместо термините «изкуствени невронни мрежи» и «изкуствен неврон» се използват по- кратките: «невронни мрежи» и «неврон». В настоящия учебник се ползват кратките термини.

Началото на моделирането чрез невронните мрежи се поставя в статията *”* *A logical calculus of the ideas immanent in nervous activity”.* Bulletin of Mathematical Biophysics на McCulloch и Pitts през 1943 г. Според тези учени типичният изкуствен неврон се представя както на **Фигура 2**. Невронът приема входни сигнали (отвън или от други неврони), които се обозначават с x0, х1, х2, ...., хn. С x0 се обозначава наличието на отклонение.

Входните сигнали могат да бъдат двоични или реални числа. При двата вида се изпълняват еднотипни изчисления и не се изисква външно управление.

Всеки от тези сигнали се умножава с определен тегловен коефициент (тегло). Той изразява силата на връзката между два неврона. Например, ако теглото е 0.5, това означава, че само половината от сигнала се предава на получаващия неврон. Ако стойността на теглото е положителна величина, връзката е възбуждаща, а когато е отрицателна – подтискаща. Множеството, съответни на входните сигнали тегла, се означава с w0, w1j, w2j, ...., wnj. Сумата от произведенията между теглата и входните сигнали се означава с net или net= **∑(xiwij),** за i- номер на входния сигнал, j- номер на получаващия неврон.Колкото е по- голям броят на събираемите в тази сума, толкова по- незначително е влиянието на отделни грешки. Това обуславя стабилността на системата от неврони.

Използването на отклонение позволява да се избегне получаването на нулева стойност за net, ако на всички входове на неврона липсва сигнал.

хо

Изход

Сумиращ

блок

Активираща

функция

w1j

w2j

wnj

:

:

wo

net

Out

х1

х2

xn

Фигура 2. Модел на изкуствен неврон (McCulloch, Pitts)

Сигналът net постъпва на входа на активираща функция и се получава изходен сигнал (Оut), наречен активност за неврона. С други думи, активността зависи от входните сигнали, теглата и от вида на активиращата (преходна) функция.

Математическият модел на **стандартния** неврон е:

Out = f(net) = f(**∑(xiwij)),**

където f е активираща функция,

xi – входни сигнали на неврона, за i от 0 до n

wij – тегла на връзките към неврона

Изходите от даден неврон към един или няколко неврона определя **структурата на връзките**.

Активността (изходът) на неврона е реално число в ограничените интервали [0,1] или [-1,1].

**Активиращата функция** представлява математическа формула, въз основа на която се определя генерираният от всеки неврон резултат.

Видът на активиращата функция определя в голяма степен възможностите на НМ и начина й на обучение.

Активиращата функция може да бъде от различен тип, но най- често е нелинейна. Най- разпространени са: прагова, сигмоидна, линейна, гаусов тип и др. Чрез нея може да се внесе **нелинейност** за представяне на зависимостта между входните и изходни величини за мрежата.

**Праговата** активираща функция е от класа на дискретните. Изходният сигнал на неврона се получава така:

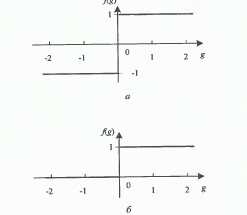
Out = -1 или 0, ако net< 0 или net=0

1, ако net >0

Математическият модел на неврон с прагова активираща функция е:

Out = sign(net) = sign (**∑(xiwij))**

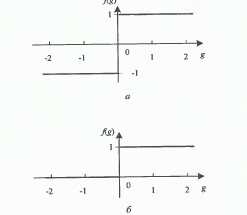
Изходът за симетричната прагова функция е в интервала [-1,1], а за смесената прагова- [0,1] - **Фигура 4**.



Out

Out

net

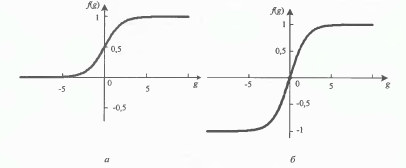
net

Фигура 4. Примерно изображение на а)симетрична и б) смесена прагова функция

Невронът със смесена прагова функция има повишена устойчивост при извеждане от работа на отделни неврони от НМ или към изменението им. Основен недостатък на праговия неврон е недостатъчната гъвкавост при обучение и настройка на НМ. Ако net незначително не достига зададения праг, то изходен сигнал не се формира. Това означава, че се губи интензивността на входния сигнал и се образува незначителен сигнал на изхода на неврона.

Невроните с прагова активираща функция не са широко разпространени.

Най- често използвана е **сигмоидната** (логистична) активиращата функция. Тя се отнася към непрекъснатите функции. Графиките на сигмоидната функция с основа експонента са показани на **Фигура** 5.



Out

Out

net

net

**Фигура 5. Смесена и симетрична сигмоидна фукция**

Формулата за изчисляване на изхода при смесената сигмоидна функция е:

1 1

Out = = ––––––––– , където e ≈ 2.7182

1+ e-net 1+ e-**∑(xiwij)**

Формулата за изчисляване на изхода при симетричната сигмоидна функция е:

1- e-net 1- e-**∑(xiwij)**

Out = = ------------ ,където e ≈ 2.7182

1+ e-net 1+ e-**∑(xiwij)**

В практиката се използват и други сигмоидни активиращи функции, но в основата им не е експонента, а арктангенс или хиперболичен тангенс.

**Линейната** активираща функция е с неограничена (Фигура 6) и ограничена (фигура 7) област на изменение. Първият вид е непрекъснато диференцируема функция, докато вторият вид има дискретен характер. Реализацията на линейната функция има невисока изчислителна сложност.

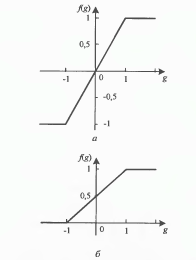


Out

net

Фигура 6. Линейната активираща функция с неограничена област на изменение

Out



Out

net

net

Фигура 7. Линейната активираща функция с ограничена област на изменение (симетрична и смесена линейна функция)

Видът на активиращата функция и нейните параметри се определя от разработващият мрежата специалист, който ги установява по експериментален път. Функцията може да бъде различна в зависимост от естеството на решаваната задача.

За изясняване на влиянието на вида на активиращата функция се разглежда следния пример.

Да се изчисли изхода от мрежата, посочена на Фигура 8, ако на входа се подава вектора [1 0.7 2.5], теглата са зададени във вектора [0.5 1.5 -1] при смесена прагова и при смесена сигмоидна активираща функция.

1

0.5

0.7

1.5

Out

2.5

-1

Фигура 8. Примерен неврон

При смесена прагова функция изчисленията се правят по условията:

0, ако net<= 0

Out =

1, ако net >0

net= 1\*0.5 + 0.7\* 1.5 + 2.5\* (-1)= -0.95

Toгава Out = 0.

При смесена сигмоидна функция:

Tъй като net e вече изчислен и е –0.95, след заместване във формулата се получава:

Out= 1/(1 + 2.7182 (0.95) ) = 0.28

Изводът е, че в зависимост от активиращата функция се получава различен изход от невроните.

**Други модели на неврони**

Показаният на Фигура 3 модел на неврон е стандартен (класически). Разработени са и нови модели на неврони- с два сумиращи блока (Паде- неврон), с квадратичен сумиращ блок, неврон с брояч на съвпадения и др.

**Паде- невронът** се състои от:

* два сумиращи блока;
* елемент, изчисляващ частното от претеглените суми (резултати от двата претеглени входа);
* активираща нелинейна функция.

Името си е получил от апроксимацията на функции по Паде метода.

Математическият модел на Паде- неврона е:

Out = f(net1/net2) = f(**∑(x1iw1ij)/ ∑(x2iw2ij)),**

където f е активираща функция, i- от 0 до n

x1i – входни сигнали на неврон към първия сумиращ блок

x2i – входни сигнали на неврон към втория сумиращ блок

w1ij – тегла на връзките към неврона към първия сумиращ блок

w2ij – тегла на връзките към неврона към втория сумиращ блок

n- брой на входните сигнали

Този неврон има два пъти повече настройващи се параметри, в сравнение със стандартния неврон.

**Невронът с брояч на съвпадения** получава на входа n- мерен двуичен вектор Х, а на изход извежда цяло число, равно на броя съвпадения на xi с wij (xi= wij).

**3. Характеристики на невронните мрежи**

Невронната мрежа наподобява мозъка поне в два важни аспекта:

* + знанията се натрупват (придобиват) в мрежата чрез процес на обучение;
  + връзките между невроните, известни като синаптични тегла, се използват за запомняне на знанията.

Теоретически, **невронните мрежи могат да моделират в неявен вид всеки процес, колкото и сложен да е той.** Тяхната сила е в способността да представят както линейни, така и **нелинейни** зависимости, както и в способността им да научават тези връзки директно от данните, които моделират. Това е особено полезно в приложения, където сложността на данните или задачата прави съставянето на такава функция по други начини неприложимо. Основно свойство на НМ е да **отразяват входната информация в изходна**.

Знанията в невронната мрежа са **неявно** представени в общата архитектура и взаимодействие между невроните. С**труктурата на връзките и теглата им са носителите на знания в невронните мрежи.** Тези знания представляват способността на мрежата да съпоставя на подадените входни данни правилните изходни резултати.

За тази цел е необходима настройка на теглата, която се постига в процеса на **обучение** на невронната мрежа. Обучението е една от основните й характеристики и за него е отделено специално внимание в следващото изложение.

Настройването на теглата на връзките към средата на работа, което е резултат от процеса на обучение, води до висока **адаптивност** на НМ.

**Невронните мрежи могат да обработват неточна и непълна или зашумена информация.** Принципно различният метод на нейната обработка в сравнение с конвенционалната обработка, осигурява тази мощна способност.

Всеки неврон приема входни сигнали от други неврони (или от външната среда) и ги използва за изчисляване на изходен сигнал. Всеки неврон работи **паралелно** с останалите **без централен контрол**. Така се реализират масивни паралелни изчисления. Обновяването на сигналите води до промяна на изхода на невроните, което може да се осъществява синхронно или асинхронно. При синхронното актуализиране всички процесорни елементи променят активностите си едновременно, докато при асинхронното- всеки възел има вероятност (обикновено фиксирана) за промяна на активността си в даден момент и най- често само един неврон може да направи това.

Невронните мрежи изпълняват **разпределена** обработка на информацията. Те се характеризират с добра производителност поради голямото количество връзки между прости изчислителни елементи.

Невронните мрежи не се влияят от малки повреди и запазват функционалността си при допустим процент дефекти в архитектурата им. Подобно явление се наблюдава и при биологичните мрежи. Всяка секунда в човешкия мозък умират средно по 3 неврона. За една година това прави около 1% загуба на неврони. Но функциите на мозъка не се нарушават, нито се редуцират възможностите му.

**4. Многослойни невронни мрежи**

Невроните в една мрежа може да са групирани в един или няколко слоя (нива). Входният слой се състои от неврони, които приемат сигнали от външната среда, а изходният- от неврони, които свързват изхода на мрежата с външната среда. Използват се междинни (латентни, скрити, вътрешни) слоеве и това са тези от тях, чиито входни и изходни сигнали остават в рамките на мрежата (фигура 9). **Скритите слоеве предстaвляват средство за изучаване структурата на данните и описване на зависимостта между изходящите и входящите неврони.** Счита се, че наличието на поне един скрит слой, е индикатор за нелинейност на изучаваната функция.



Легенда: xi- входни източници, Outi- изходни сигнали

Фиг.9. Многослойна невронна мрежа

В научната литература няма установени правила за дефиниране на оптималния брой на скритите нива. Най- често той се определя чрез експерименти.

Изходните неврони получават преработените сигнали от предходните нива и генерират отговора на системата. **Като правило отговорът на НМ е числов вектор.** Ако за решаването на една задача трябва да се получи резултат, например в градуси, то е необходимо да се преобразува изходния вектор.

Връзките между невроните по слоеве може да бъдат представени чрез двумерна матрица **Wi** (фигура 10): матрица W1 - между невроните от първо и второ ниво, а матрица W2 – между второ и трето ниво. Tеглата на връзките от неврон i към неврон j от следващ слой са зададени чрез **wij**. Например:

1

-0.6

1.3

1

1

1.2

2

2

-0.8

-1.7

2

3

3

4

0.1

Трето ниво

Второ ниво

Първо ниво

- 0.6 1.2 -1.7 1.3 -0.8

W1 = ............................ W2= ………….

........................... ………….

.....................0.1

Фиг. 10. Матрици, които описват теглата на връзките в невронна мрежа

За да се определи **архитектурата** на невронната мрежа трябва да се зададе топологията й и активационните функции на невроните.

**5.Видове НМ според топологията**

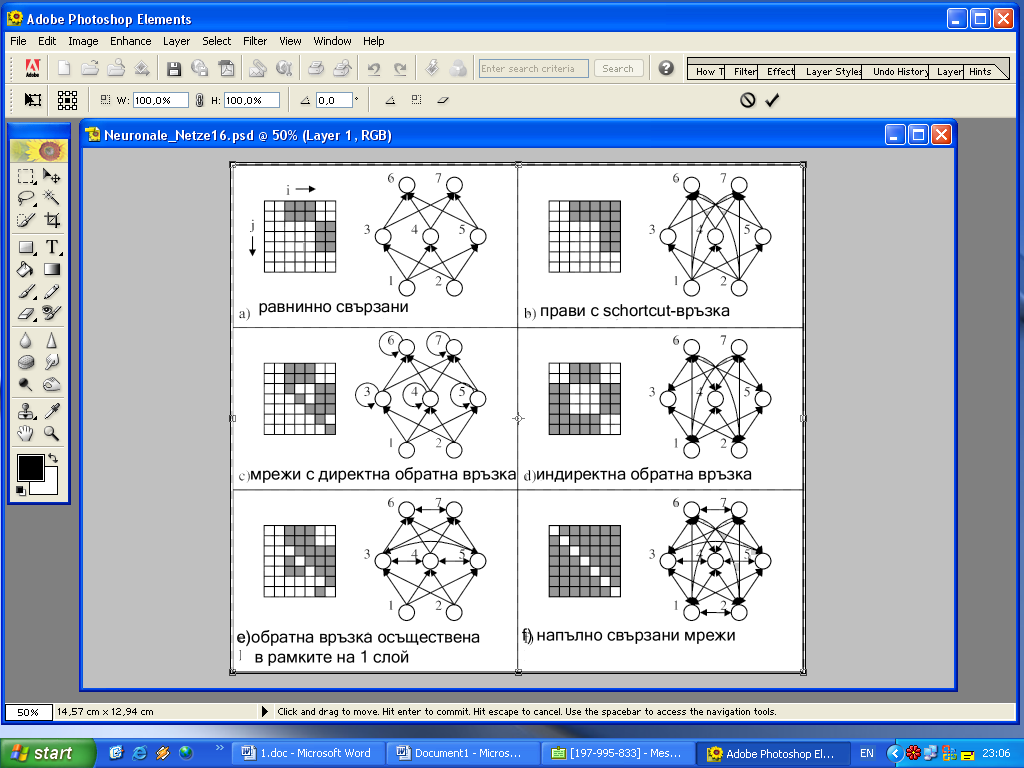
Разположението и начинът на свързване между невроните определят **топологията** на мрежата. Топологията на невронната мрежа и теглата на връзките между невроните определят функционалността й.

В процеса на обучение топологията на невронната мрежа се модифицира. Така тя се настройва за съответствие към задачата. Аналогично, невроните в мозъка умират и се създават нови синаптични връзки в процеса на съществуване на индивидите. Примери за успешно използвани топологии при сравнително големи мрежи са 203 (брой на входните невроните) – 80 (брой на невроните на междинно ниво) – 26 (брой на изходните неврони), 960 – 9 – 45, 459 – 24 – 24 – 1 и др.п.

Посоката на връзките в невронните мрежи може да бъде от входния към скрития слой и от скрития към изходния и тогава тези мрежи се наричат **прави (**с право разпространение на сигнала)**.** При тях няма връзки, излизащи от изхода на невроните от един слой и завършващи на входа на неврони от същия или предишни слоеве. Повечето невронни мрежи с право разпространение са **пълносвързани**, т.е. входящите неврони са свързани единствено с всички неврони от следващото ниво, а те на свой ред – с всички възли от следващия слой. Ако част от синаптичните връзки отсъстват, мрежата е **непълносвързана**.

Различават се два вида прави НМ:

* Равнинно свързани- връзките са само между неврони от 1 слой към непосредствено след него намиращия се слой (фиг.11- а). Toва са най- често използваните НМ.



Фигура 11. Основни разновидности прави и рекурентни мрежи

* Прави с shortcut връзка- връзките между невроните от слой k към неврони от слой k+i, където i>1, т.е връзката е по посока от входа към изхода, като се прескача един или няколко слоя (фиг. 11 b);

Когато връзката е насочена от изхода на един неврон към входа на друг, мрежите се наричат **рекурентни** (с обратна връзка). При тях е възможна и връзка между невроните в рамките на един слой. За всяка рекурентна НМ може да се построи идентична мрежа с право разпространение на сигнала. Смята се, че човешкият мозък повече прилича на рекурентна мрежа, отколкото на права.

При тези мрежи има временна задръжка на сигнала, т.е. той не се разпространява мигновено. Тъй като активацията се връща към породилите я неврони, тези мрежи имат вътрешни състояния, запомнени в активиращите нива на процесорните единици. Съществени за тези мрежи са именно техните динамични свойства. Примери за рекурентни мрежи са мрежите на Hopfield (фигура 12), машините на Boltzmann и др.



Фиг.12 Мрежа на Hopfield

Различават се няколко типа рекурентни мрежи:

* *С директна обратна връзка* - при тези мрежи изходът на даден неврон се явява като вход на същия, като по този начин невроните могат сами да се възбуждат (активират)- фиг.11 - с;
* *С индиректна обратна връзка -* посоката на връзките е от неврони от по- висок слой към неврони от по- нисък- фиг.11- d;
* Мрежи, при които обратната връзка се осъществява в рамките на 1 слой – невроните от 1 слой са свързани помежду си. Този вид връзка се използва при т.нар мрежи „победителят взема всичко”, където само един неврон (този, който е най- силно активиран) от слоя остава активен, а другите се деактивират в даден момент от времето – фиг.11- e.
* *Напълно свързани мрежи без директни обратни връзки –* съществуват връзки между всички неврони- фиг.11- f*.*

**6.Перцептрони**

Перцептроните са най- простата форма на права невронна мрежа, използвана за класификация на образци. Първите са създадени от Frank Rosenblatt през 1957 г., а в последствие са разработени различни варианти на перцептрони. Един от базовите модели е мрежа с един неврон (**Фигура 13**). Теглата и отклонението му могат да се настройват, за да произвеждат коректен на входния вектор от сигнали изход.

**∑**

Активи-раща функция

xo

x1

x2

xi

.

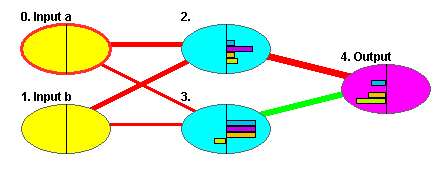
.

**Out**

Фигура 13. Схематично изображение на перцептрон

Такъв перцептрон е ограничен за разпознаване на образци само от 2 класа. С разширяването на мрежата, с включване на междинни слоеве, може да се правят класификации на повече образци.

Така се създава многослойния перцептрон Фигура 14, който е права мрежа и е най- често използвания вид днес.



Фигура 14. Перцептрон с междинен слой за реализация на XOR

С многослойна невронна мрежа може да бъдат представени булевите функции. Всяка от тях може да се изрази като дизюнкция от конюнкции. Конюнкциите се реализират на междинно ниво, след което се обединяват чрез дизюнкция.

Доказва се (Robert Necht - Nilsen, 1986 г.), че всяка нелинейна функция, може да се представи чрез невронна мрежа с не повече от 2 междинни слоя. Доказателството е основано на почти идентичната теорема на Андрей Колмогоров, известна във функционалния анализ. Едно неудобство на това доказателство е, че за това представяне са необходими произволен брой неврони в първия междинен слой. По- късно Funahashi (1988 г.) и Сибенко (1989 г.) доказват независимо един от друг, че е възможно представяне чрез краен брой неврони в двата междинни слоя и че даже за някои “гладки” функции един междинен слой е достатъчен. Тези доказателства са емпирично потвърдени от множество приложения на невронни мрежи, в които се използва един междинен слой.

Многослойните перцептрони имат някои отличителни признаци:

* Всеки неврон има нелинейна активационна функция, като най- често използвана е сигмоидната функция. Наличието на нелинейност е много важно, тъй като именно чрез нея се представят по- сложни връзки между входа и изхода, а не просто отразяване на входните данни в изходни (както е при еднослойния перцептрон).
* Мрежата съдържа един или няколко скрити слоя неврони. Те позволяват обучението за сложни задачи, извличайки най- важните признаци от входния вектор.
* Мрежата притежава висока степен на свързаност между невроните.

Комбинирането на тези признаци, заедно със способността да се обучава от опита си, води до високата изчислителна мощ на многослойният перцептрон.

**7. Методи за предварителна обработка на данните**

Входните данни, които се подават на НМ, може да са текстови, графични, числови, дати и др. Но **НМ може да обработи само числа**. Преди да се подадат на входа на мрежата различните типове данни трябва да се преобразуват. В различните софтуерни продукти за невронни мрежи се прилагат разнообразни методи за трансформация.

Често използван метод е мащабирането. То се прави с цел изравняване на диапазона на изменение на величините. Прилага се следната формула за претегляне на числовите стойности:

x – x min

xm = ------------- (b-a) + a

xmax - xmin

x- действителната стойност на подадената стойност

хm- мащабирана стойност на подадената стойност

хmin, xmax –реален диапазон на изменение на стойността

а,b- приемлив диапазон на изменение на входните стойностите

Отговорите на мрежата следва да бъдат преобразувани, обратно на мащабирането.

Outm – a

Out = ------------- (Outmax - Outmin) + Outmin

b - a

Out – реалният изход от неврона

Outm- мащабираната получена стойност

Outmax , Outmin - реален диапазон наполучени стойности в мрежата

а,b- приемлив диапазон на изменение на стойностите

В софтуерния продукт Alyuda NeuroIntelligence колоните, съдържащи категории се кодират по следните начини: един-от-N, двоично и числено кодиране. Един-от-N кодиране означава, че колоната с N отделни категории се кодира в множество от N числови колони, с една колона за всяка категория. Например, стойности като „нисък”, „среден” и „висок”, ще бъдат представени съответно като (1,0,0), (0,1,0) и (0,0,1). Двоично кодиране означава, че колона с N различни категории се кодира в множество от M двоични колони, където M е равно на дължината на бинарното число, необходимо за представяне на N различни стойности. Например, стойности „червено”, „жълто”, „зелено”, „синьо” и „бяло”, ще бъдат представени съответно като {1,0,0}, {0,1,1}, {0,1,0}, {0,0,1} и {0,0,0}. Числено кодиране означава, че колона с N отделни категории се кодира в една числова колона с една целочислена стойност за всяка категория. Например, стойности „нисък”, „среден” и „висок”, ще бъдат представени съответно като {1}, {2} и {3}.

В друг софтуерен продукт за невронни мрежи- EasyNN [29], всеки текстов стринг има асоциирана числова стойност- сума от произведенията на ASCII кода на всеки символ и позицията му в стринга.

**8.Методи за интерпретиране на отговорите на НМ**

Най- често използваният метод за интерпретиране на отговорите на НМ е **„победителят взема всичко”.** Обикновено се използва в задачи за класификация. При тях е необходимо да се определи принадлежността на всеки вектор към определен клас. Номерът на изходния неврон с максимална стойност се интерпретира като номер на класа, към който се отнася подадения вектор (пример).

Друг метод е **знаковата интерпретация**. Изисква се броят на невроните да бъде log2M, където M е броят на класовете. Ако се обозначи съвкупността от изходни сигнали с y1, y2, …., ym и се заменят положителните сигнали с единици, а отрицателните с нули, получената двуична последователност се разглежда като номер на разпознатия клас в двоична бройна система.

Посочените методи не изчерпват всички възможни подходи. За всяка предметна област е необходимо експерименталното им изследване.

1. **Обучение на невронни мрежи**

Сред основните характеристики на невронните мрежи, от особено значение е способността им да се учат от своята среда и да подобряват работата си чрез обучение.

Обучението е процес, при който се настройват теглата и връзките между невроните чрез моделиране на средата, в която работи мрежата.

По време на обучението мрежата автоматично извлича функционални зависимости от входните данни. В края на процеса тези зависимости се запомнят като теглови стойности и структура на връзките между невроните и по- късно се използват. Съществуват алгоритми, които позволяват мрежата да се обучава автоматично. Но няма универсален алгоритъм, подходящ за всички видове мрежи.

Нека да припомним, че биологичните невронни мрежи също се обучават, защото не се раждат с цялото знание, прилагано по време на съществуването им. Тяхното обучение изисква известен период от време, през който биологичната мрежа са модифицира така, че да приема и съхранява нова информация. Основите на теорията на обучението са поставени от J.Hebb през 1949 г. в книгата “Организация на поведението” (”Organization of Behaviour”).

За да се получи достоверен резултат след обучението на невронната мрежа, са необходими подбрани примери по проблема. Те представляват комбинации от конкретни реални стойности за наблюдаваните величини и съответния им резултат.

Обучението се реализира чрез повтарящ се итеративен процес на промяна в тегловните коефициенти на връзките между невроните, до тогава докато се изпълни зададен критерий. Той може да бъде определено ниво на грешката, брой епохи или друг. Въздействията на входа, които невронната мрежа получава по- късно, могат да се различават от сигналите, с които тя е обучена, но получената реакция трябва да бъде правилна.

**Обработката на знания** в многослойни мрежи се състои от обмен на сигнали между слоевете неврони в мрежата.

Обучението на невронни мрежи включва следната последователност от действия:

* + невронната мрежа е стимулирана от средата,
  + променят се всички тегла. Типът на алгоритъма определя начина, по който става това.
  + в резултат на промените в структурата на НМ, тя отговаря по нов начин на входните въздействия.

Процесът продължава, докато се изпълни поставеният критерий.

Когато обучаващият процес приключи, мрежата е готова за използване. Постига се стабилно състояние, т.нар. **конвергенция**. Обикновено при експлоатацията теглата не се променят, докато мрежата се използва за задачата, за която е обучена.

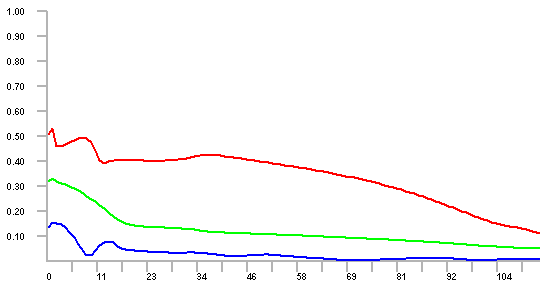
Като резултат от обучението мрежата „научава” връзката между входящите и изходящи величини и да обобщава.

**Обобщение е способността на мрежата не само да се учи от обучаващото множество, но и да показва добри резултати с нови данни.**

В това се състои фундаменталната разлика между НМ и традиционните модели за обработка на информация. При последните се създава модел на средата, който се проверява с реалните данни. Създаването на НМ се основава директно на реалните данни. Така НМ осигуряват не само модел на средата, но и реализират обработка на данните.

Ефективността на обучението се определя чрез сравнение между получения от мрежата резултат и очаквания (желания). Тази разлика дава грешката. Тя трябва да намалява в процеса на обучение.

В процеса на обучение може да се следи промяната на графиката на грешката. **За всеки неврон тя представлява разликата между получената и желаната стойност за всеки подаден пример**. Наблюдава се кумулативната грешка за целия набор от данни (**Фигура 15**).



Грешка

Цикли

Локални минимуми

Фигура 15. Графики на грешки (в реда от най- близката до абсцисата нагоре- минимална, средна и максимална) в примерна невронна мрежа

Невронните мрежи се обучават чрез модификация на структурата и теглата на връзките между невроните като резултат от получената отвън информация.

Данните трябва да се проверяват за коректност. Възможно е да се открият грешки при въвеждане или да липсват някои стойности. Може да има данни, които са извън областта на останалите примери. Не винаги това са грешки. Някои от тези данни могат да са носители на полезна информация (например за повреда на машина). Разумно е първо да се приложи статистически анализ за идентификация на потенциални проблеми. Ако се установят погрешни данни, се препоръчва, те да се отстранят от обучаващото множество.

Обикновено подбраните данни се разделят в **три групи:**

* Извадка за **обучение** (тренировъчно множество), състояща се от примери (наблюдения), въз основа на които се извършва обучението на невронната мрежа.
* **Валидираща** извадка, която служи за проверка на възможностите на обучените невронни мрежи да дават правилен отговор при заявка и за избор на конкретна архитектура между множество алтернативни архитектури, като условие за приключване на обучението и др. Използва се за настройване на мрежовата топология или други параметри на мрежата, с изключение на теглата.
* **Тестова** извадка, която обхваща новите примери за определяне до каква степен се е обучила мрежата, когато трябва например да се класифицират нови данни. Тя се използва за проверка и разчитане на грешките на мрежата. Чрез нея може да се установи дали мрежата действително извършва обобщение.

Пропорцията между извадките се основава на експерименти. Например 70% обучаващи и 30% тестови примери, 80% обучаващи и 20% тестови, 90% обучаващи и 10% тестови.

Еднократното преминаване на цялото обучаващо множество през мрежата се нарича **епоха**.

Подготовката на невронната мрежа включва и избор на **критерий за преустановяване на обучението**. В качеството на такъв критерий може да се използва предварително зададено ниво на грешка, което крие риск от доста продължително обучение при по- сложни архитектури или определен брой итерации/епохи или при регистриране на ръст на грешката във валидиращото множество и др.

За „свръхобучение” (преобучение) може да се говори, когато невронната мрежа демонстрира добри резултати в обучаващото, но не и в тестовото множество (не може да обобщава). Ако НМ се обучава с много голям брой примери, тя може да запомни данните от обучаващото множество. НМ намира признаци, които присъстват в обучаващото множество (поради шум), но не са характерни за изучаваната функция. Когато невронната мрежа е „преобучена”, тя губи способността да обобщава аналогични входни сигнали.

Ако при тестовите данни грешката намалява, тогава системата извършва обобщение, но ако грешката на обучаващите данни продължава да намалява, а грешката на тестващите данни се увеличава, то мрежата е спряла да обобщава и просто запомня обучените данни. В такива случаи обучението обикновено се прекратява.

При многослойния перцептрон се счита че:

* Локалните признаци в данните се извличат в първия скрит слой, т.е. някои неврони от първия скрит слой може да се използват за разделяне на входното пространство на различни области, а останалите неврони в слоя да се обучават на локалните признаци, характеризиращи областта.
* Глобалните признаци в данните се извличат във втория скрит слой. Невроните от втория слой обобщават входните сигнали от невроните от първия скрит слой, отнасящи се към конкретна област на входното пространство. Така те се обучават на глобалните признаци в тази област.

**10. Класификация на основните подходи за обучение на НМ.**

1.Обучението може да протича off- line, on- line или комбинирано. В първия случай невронната мрежа се обучава чрез множество данни, разпространявано многократно през мрежата. При всяко преминаване последната оптимизира структурата си, за да достигне зададените (целевите) стойности. При off- line – фазата на обучение е отделена от фазата на използване на невронната мрежа. Много традиционни обучаващи алгоритми (като backpropagation) използват този тип обучение.

On- line обучението се разглежда като научаване на всеки пример по отделно, докато мрежата работи и данните съществуват за кратък период от време. След всеки пример системата прави промени в своята структура за оптимизиране на целевата функция. Типичен случай за този вид обучение е, когато данните постъпват произволно от средата и се зареждат в мрежата едни след други. Съществува възможност eдни и същи примери да се разглеждат няколко пъти, което се приема като нетипичен случай, в контраст с off- line обучението, където един пример се представя в системата много пъти като част от процедурата на обучение. Методи на on- line обучение са публикувани в трудовете на Albus (1975), Fritzke (1995), Saad (1999) и др.

Методите за on- line обучение са адаптивни обучаващи методи. Адаптивното обучение е типично за много биологични системи и е също полезно в инженерните приложения, роботиката, контрол на процесите и др. Значителни постижения в този вид обучение са отбелязани в адаптивната резонансна теория (Carpenter, Grossberg- 1990, 1991).

При комбинираното on- line и off- line обучение, системата може да работи първоначално в on -line режим, след което да превключи в off- line режим и т.н. Използва се за оптимизация, чрез “хващане на малък прозорец” от данни от непрекъснат входен поток.

1. Според това дали в процеса на обучение се използва информация за коректността на резултата, основните типове обучение са четири:с учител, подпомагано, без учител и комбинирано.

Най- разпространената форма на обучение на НМ е обучението с учител **(надзиравано обучение)**. На невронната мрежа се подава обучаващо множество от данни, съдържащо желания (целевия) изход за всеки входен сигнал. Желаният резултат представлява оптималните действия, които мрежата трябва да изпълни. Например, данните може да се разглеждат като двойки (x,y), като x e входа, а y коректният изход. Параметрите на мрежата се коригират с отчитане на грешката- разликата между фактически получения и желания резултат. Корекциите на параметрите се извършват постъпково, с което се имитира обучението с учител. След приключване на обучението мрежата може да работи самостоятелно «без намесата на учителя». Пример за този вид обучение е алгоритъмът с обратно разпространение на грешката (backpropagation).

При **подпомаганото обучение** (невродинамично програмиране) се комбинира динамичното програмиране и обучението с учител. Динамичното програмиране (Belman, 1957) реализира математически формализъм на последователно вземане на решения. Ако виртуално се «премести» подпомаганото обучение в областта на динамичното програмиране, може да се ползват всички резултати от изследванията му. Процесът за вземане на решения съответства на Марковския модел.

НМ се обучава в два главни аспекта: на основа на наблюдения на собственото си поведение да взема добри решения и да ги подобрява с помощта на механизма на подкрепа.

На невронната мрежа се подава целта, която трябва да постигне. Обучението протича чрез итеративни взаимодействия със средата на НМ от тип «проба- грешка». За всеки изходен сигнал, невронната мрежа получава съвсем малко информация дали той е правилен или грешен. Това обучение се смята за психологически по- обосновано, защото обикновено човек не получава пълен образец на желаното поведение.

Методите на невродинамичното програмиране се използват при задачи с големи пространства от състояния, където на първо място стои планирането.

При обучението без учител(самообучение) съществува единствено независима от задачата мярка за качеството на представяне на функцията в НМ и свободни параметри (тегла и структура на връзките), които се оптимизират по отношение на мярката. Теглата на връзките се настройват така, че представянето на данните в мрежата да е най- добро, съгласно зададения критерий за качество на представянето. За разлика от обучението с учител априори не съществува множество от категории, към които да се класифицират входните образци. НМ трябва сама да изработи свое представяне на входните сигнали. По време на процеса на обучение, невронната мрежа сама „открива” групите, в които трябва да бъдат класифицирани входните образци.

Самообучението може да бъде полезно, само когато съществува изобилие от входни данни. Без него би било невъз­можно да се намерят някакви характеристики на данните, които обикновено изглеждат като случаен шум. В този случай изобилието дава знание. Пример за обучение без учител е конкурентното или състезателно обучение.

Обикновено обучението с учител се извършва оff- line, докато това без учител – оn- line.

Съществуват и НМ с комбинирано обучение, т.е. съдържат сегменти от обучение с учител и без учител обучение. В тях частта „без учител” се обучава отделно.

3. Обикновено процесът на обучение се изпълнява само веднъж върху цялото множество от данни (понякога това може да отнеме много итерации). След това системата е готова за използване и не се тренира отново. Този тип обучение е най- разпространеният в съществуващите модели. Но съществува и друг подход, при който невронната мрежа се учи непрекъснато от постъпващи данни в променящата се среда през цялото време на съществуването си и това е т.нар. обучение през целия живот.

4. Според наличието на промени в броя на невроните и връзките има два вида невронни мрежи: с фиксиран размер и с динамично променяща се структура.

5. Хората са селективни в обработката на информация. Те активно търсят само нова информация. Аналогично при невронните мрежи се разглеждат два типа обучение: **активно**- с подбор на данни, филтриране и търсене на релевантни данни, и **пасивно**- системата приема всички постъпващи данни.

Много изследователи се опитват да определят капацитета на една невронна мрежа, т.е. максималния брой образци, които могат да се запомнят и разпознават стабилно от мрежата. Според Naykin, 1994 г. броят им е:

N

p max = -------

1. ln N ,където N е броят на невроните в мрежата

**11.Алгоритъм с обратно разпространение на грешката**

Един от най- изучаваните алгоритми е алгоритъмът с обратно разпространение на грешката **(**backpropagation). Използва се за обучение на многослойни перцептрони и представлява приложение на делта правилото.

**Backpropagation** е в основата на много разработки в областта на невронните мрежи. Откриван е няколко пъти независимо един от друг от Werbos (’86), Parkar (’85) и Rumelhart, Hinton и Williams (’86). Алгоритъмът дава предписание за промяна на теглата във всяка права мрежа, която трябва да научи обучаващо множество от образци.

Алгоритъмът се изпълнява на 2 етапа (паса). На входа на мрежата се подават сигнали и при преминаването им в права посока през мрежата (първия пас) се получават изходни сигнали от мрежата. При обратния пас, след сравняване на получения изход с целевите стойности (правилния изход), се корегират теглата на връзките между невроните от n+1 слой и невроните от n -тия слой (фиг.16).

Първи етап: Разпространение на сигналите от входа към изхода

Изхо-ден слой

Скрит слой

Входен слой

Втори етап: Обратно разпространение на грешката

Фигура 16. Обща идея на алгоритъма Backpropagation

Критериите за преустановяване на обучението могат да бъдат различни. Често използван е критерият за достатъчно малка промяна на средноквадратичната грешка в рамките на една епоха. Промяната се счита за достатъчно малка, ако е в интервала 0.1 – 1% за епоха.

Друг критерий е след всяка итерация да се тества способността на мрежата за обобщение. Когато се постигне приемлива грешка на обобщение, обучението спира.

За пълното обучение на мрежата са необходими различен брой цикли (**Фигура 17** ).

Пресмята-не на изхода на мрежата

Текущият изход се сравнява с целевия

Въвеждат се валидиращи данни

Пресмята се изхода

Определя се грешката

Промяна на параметрите на мрежата

Промяна на теглата на връзките

Въвеждане на обучаващи данни

Определяне на архитектурата

Достигнат желан резултат

Грешката е твърде голяма

Отклонението е мн.голямо

Достигнат желан резултат

Фигура 17. Основни стъпки в процеса на обучение

Проблемите при разглеждания алгоритъм са свързани с ниската скорост на обучение на невронната мрежа. Причината е, че установяването на теглата в един слой зависи от теглата във всички следващи слоеве. Това може да бъде избегнато до известна степен чрез ненадзиравано обучение (самообучение) или чрез хибриден подход, при който някои слоеве се самообучават единствено от изходите от пре­дишния слой.

Невронните мрежи, обучавани по този алгоритъм са широко разпространени.

Методът на обучение- backpropagation, спада към т.нар. **детерминирани** методи. При тях стъпка по стъпка се коригират теглата на мрежата, чрез използване на техните текущи значения на входовете, фактическите и желаните изходи.

Целта на алгоритъма на обратно разпространение е намиране на глобалния минимум на функцията на грешката. Поради начина, по който грешките се „разпространяват обратно” по мрежата, НМ може да открие **локален минимум**, вместо глобален. Мрежата няма да е достатъчно обучена и теглата няма да са добре настроени, а ще се получи стабилно състояние. Това е състояние, в което никой изход от неврон не се променя повече. Локалният минимум може да бъде близо, но не е оптималното решение. Сигнал за възникването на подобен проблем е продължителното обучение на невронната мрежа. Друг индикатор е динамиката на грешката по време на обучението. Първоначално спада, след което се уравновесява или се колебае. Или, ако тя не намалява с повече от 1% от предишната си стойност за 100 корекции на теглата, то считаме, че сме попаднали в локален минимум.

Специализираната научна литература изобилства от подходи за преодоляване на този недостатък на алгоритъма. Ефективен начин за избягване на риска от попадането на локален оптимум е използването на инерционен параметър (Мomentum в EasyNN). Предназначението на инерционния параметър е да гарантира същата посока на изменение в тегловните коефициенти както при последната промяна. Когато не се използва, резултатите на НМ са чувствителни към избора на стойност на скорост на обучение (Learning rate в EasyNN - η). Ниските стойности на η  водят до „забавяне” процеса на обучение, а високите предизвикват колебания в стойностите на тегловните коефициенти. Включването на инерционен параметър  в алгоритъма на обучение позволява при по- високи стойности на η, водещи до ускоряване процеса на обучение, да се минимизира риска от колебание в теглата.

Подходящ метод за отдалечаване от даден локален минимум е да се добавят към всички тегла случайни малки стойности. Така процесът на обучение продължава от друга точка.

Друг изход е да се добави един скрит слой неврони към невронната мрежа. Процесът на обучение продължава, тъй като се променя размерността на пространството на теглата и следователно- на повърхнината на грешката в него.

**Пример:** Да се класифицират (разпознаят) цифрите от 0 до 9 в съответствие със сигналите, подавани на входа на невронна мрежа, обучена по метода backpropagation. Тъй като трябва да се разпознаят девет цифри (без да се брои нулата), в целевия вектор на изход следва да има девет неврона. Ако се получи нула във всички позиции, се счита, че е разпозната нула. Така, че за нея не се отделя специален неврон. Например целевият вектор за 2 е [ 0 1 0 0 0 0 0 0 0]. Вярната цифра се отбелязва с 1 в съответната позиция.

Нека започнем с описание на цифрата 1. Решетката, изобразяваща цифрата, е показана на фиг.18 .

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |

Фигура 18. Графично представяне на пикселите за цифрата 1

Числата, които ще се разпознават, предварително се представят чрез решетка 9 х 7, като светлите пиксели са нули, а черните – единици. Битовете се представят, започвайки от горния ляв ъгъл на решетката, колона по колона.

Единият от примерите, с които ще се обучи мрежата за разпознаване на 1, се задава по следния начин:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Стойност** | **Вход** | **Стойност** | **Вход** | **Стойност** | **Вход** | **Стойност** |
| vh1 | **0** | vh10 | **0** | vh19 | **0** | vh28 | **1** |
| vh2 | **0** | vh11 | **0** | vh20 | **1** | vh29 | **1** |
| vh3 | **0** | vh12 | **0** | vh21 | **0** | vh30 | **1** |
| vh4 | **0** | vh13 | **0** | vh22 | **0** | vh31 | **1** |
| vh5 | **0** | vh14 | **0** | vh23 | **0** | vh32 | **1** |
| vh6 | **0** | vh15 | **0** | vh24 | **0** | vh33 | **1** |
| vh7 | **0** | vh16 | **0** | vh25 | **0** | vh34 | **1** |
| vh8 | **0** | vh17 | **0** | vh26 | **0** | vh35 | **1** |
| vh9 | **0** | vh18 | **0** | vh27 | **1** | vh36 | **1** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Стойност** | **Вход** | **Стойност** | **Вход** | **Стойност** | **Изход** | **Стойност** |
| vh37 | **0** | vh46 | **0** | vh55 | **0** | **izhod\_za\_1** | **1** |
| vh38 | **0** | vh47 | **0** | vh56 | **0** | **izhod\_za\_2** | **0** |
| vh39 | **0** | vh48 | **0** | vh57 | **0** | **izhod\_za\_3** | **0** |
| vh40 | **0** | vh49 | **0** | vh58 | **0** | **izhod\_za\_4** | **0** |
| vh41 | **0** | vh50 | **0** | vh59 | **0** | **izhod\_za\_5** | **0** |
| vh42 | **0** | vh51 | **0** | vh60 | **0** | **izhod\_za\_6** | **0** |
| vh43 | **0** | vh52 | **0** | vh61 | **0** | **izhod\_za\_7** | **0** |
| vh44 | **0** | vh53 | **0** | vh62 | **0** | **izhod\_za\_8** | **0** |
| vh45 | **1** | vh54 | **0** | vh63 | **0** | **izhod\_za\_9** | **0** |

Задават се по няколко примера за обучение за разпознаване на една и съща цифра. В тях могат някои от черните пиксели да бъдат заменени със светли и обратно,т.е. да бъдат подадени „зашумени” данни (фигура 19).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |

Фигура 19. „Зашумено” представяне на единицата

По аналогичен начин се работи за задаване на останалите цифри.

Подготвя се архитектура на невронна мрежа от следния тип 63-6-9. На входа на всеки от тези 63 неврона се подава сигнал единица или нула, в съответствие с решетката, изобразяваща цифрата, или като цяло на вход се подава стринг от битове от единици и нули.

Изходният неврон се разглежда като активен, ако активацията му е по- голяма от 0.9 и като неактивен, ако активацията му е под 0.1.

След обучението, невронната мрежа разпознава коректно цифрата.

Проблеми възникват при разпознаване на цифрите 8 и 6 при зашумено подаване на 8. Получава се активация на шестия изходен неврон 0.53, а на осмия- 0.41. Тези стойности не са достатъчни за категоричното определяне на цифрата на изход.

Едно от възможните решения е да се направи по- прецизна решетка върху цифрите, с по- голям брой пиксели по вертикала и хоризонтала. Това обаче, ще увеличи времето за обучение на мрежата.

По подобен начин може да се реши проблема за разпознаване на букви.

**12.Основни предимства и приложения на невронните мрежи**

**Предимствата** на невронните мрежи могат да се обобщят така:

1. Паралелизъм на работа. За разлика от последователното изпълнение на инструкции при конвенционалния компютър, моделите на невронните мрежи използват паралелизма на многопроцесорната система. Това позволява значително ускоряване на обработката.

2. Адаптивност. Те са приспособими към всякакъв вид задачи след подходящо обучение.

3. Способност за обобщение. Невронните мрежи могат да извличат зависимости от множество данни. Невронната мрежа може да се ползва за изграждане на модел, който открива връзки между входовете и търсения изход. Това покрива много широка област от проблеми.

Някои **недостатъци** на невронните мрежи накратко са:

- за сложни задачи се изисква голям брой неврони, което трудно се реализира технически;

- при големите невронни мрежи обучението е много бавно;

- трудности при генериране на обяснение на решението;

- проблеми при изграждане на естествено- езиковия интерфейс.

**13.Типове задачи, решавани с НМ**

Основните групи задачи, за които е подходящо прилагането на невронните мрежи са **класификация и разпознаване на образи, разпознаване на реч и на ръкописен текст, диагностиране, прогнозиране** и др.

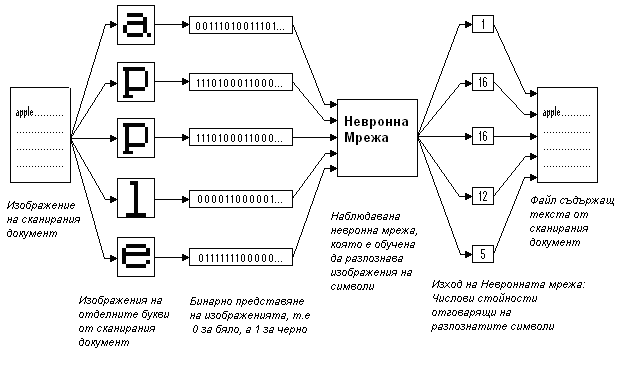
Невронната мрежа, тренирана за **класификация**, определя към коя група принадлежат подадените входни значения. Тези групи могат да бъдат, както с размити, така и с ясно определени граници.

Успешни са приложенията за класификация на акции, за автоматична оценка на фирмен или частен кредитен рейтинг на основа финансови показатели и др.

**Разпознаването** на **образци** е една от най- често срещаните употреби на невронни мрежи. Представлява форма на класификация. На обучената невронната мрежа се представя образец. Това може да бъде изображение, звук или какъвто и да е вид данни. След това невронната мрежа се опитва да определи дали входните данни съвпадат с образеца, който е била тренирана да разпознава. Той трябва да бъде разпознат дори ако подадените входни данни са зашумени.

Невронните мрежи работят добре при задачи за **разпознаване на реч** и на ръкописен текст.

Част от софтуера за **разпознаване на текст** използва невронни мрежи, в частност перцептрони, като класификационен механизъм. Както е известно този софтуер позволява да се сканира печатан текст, след което да се конвертира във файл с формат на Word документ, където може да се обработва. В този процес на конвертиране, софтуерът трябва да анализира всяка група пиксели, които изграждат една буква и да даде резултат, кореспондиращ на знака (**Фигура** 20).



Фигура 20. Конвертиране на сигнали от сканиран документ в текст чрез НМ

НМ се прилагат успешно за **диагностика**:

-машинна. Невронните мрежи служат за установяване на причините за спирането на машините и предотвратяването на тези проблеми;

-медицинска. Невронните мрежи се използват за асистиране на лекари чрез анализиране на симптоми и/или образ.

Прогнозирането е друго разпространено приложение на невронните мрежи. При подаване на входни данни под формата на времеви редове и с последователно отместване на фиксиран „прозорец”, невронната мрежа може да прогнозира бъдещи стойности.

Подходящо е прогнозиране на пазара на акции, пазарни показатели, борсови цени, банкрути и др. чрез използване на фирмени данни; маркетингови прогнози; метеорологични прогнози и др.

**14.Практически приложения на НМ**

1. НМ са универсални апроксиматори, тъй като са в състояние да изгладят различен тип функционални зависимости, независимо от степента им на сложност и линейност/нелинейност. Това ги прави подходящи за решаването на редица практически проблеми, тъй като обикновено е по- лесно да се намерят данни, отколкото да се направи коректно предположение за основните „закони”, управляващи системата, от която са генерирани тези данни.
2. В тясна връзка с предходното приложение е използването на НМ в технологиите Data mining (DM).
3. Контрол:

- на качеството на изделия. Чрез добавяне на камера или сензор в края на производствения процес, автоматично могат да се инспектират дефекти.

- на кредитни картови транзакции и автоматично спиране на разплащането при засичане на измами.

- на автопилотиране на автомобили.

Всяко приложение се изисква специална архитектура за невронната мрежа. В нея трябва да се вграждат повече знания за проблема при подходящо намаляване броя на връзките.

1. Създаване на неврокомпютри.

Интересът към НМ от инженерна гледна точка е провокиран най- вече от това, че невронни мрежи, подобни на многослойните перцпетрони, имат потенциал за създаване на универсални изчислителни машини. Те предоставят алтернативна форма на обработка на информацията, която се оказва по- подходяща за решаването на редица типове задачи.

Основни структурни компоненти на неврокомпютъра (фигура 21) са:

- конструктор на проекта на НМ;

* невроимитатор;
* препроцесор;
* база данни;
* интерпретатор;
* интерфейси за вход/изход.

В състава на невроимитатора се включват: невронна мрежа, алгоритми за обучение на невронната мрежа, алгоритми за формиране на топологията й, блок за избор на информативни признаци и блок за оценка.

Блок за избор на информат. признаци

Алгоритми за формиране на топологията на НМ

Невронна мрежа

Интерпретатор

н

е

в

р

о

и

м

и

т

а

т

о

р

Интерфейс за извеждане

Интерпретатор

Алгоритми за обучение

Блок за оценка

База от данни

Препроцесор

Интерфейс за вход

К

О

Н

С

Т

Р

У

К

Т

О

Р

на

П

Р

О

Е

К

Т

А

на

НМ

Фигура 21. Схема на неврокомпютър

Алгоритмите за формиране на топологията на НМ са предназначени за създаване на мрежа, съответна на модела на предметната област. Тъй като няма формална процедура за определянето й, се използват евристични процедури. Те се основават на опита за работа с аналогични невромрежови модели за същата предметна област. Ако такъв опит няма, изборът на архитектура е чрез претърсване на много възможни варианти на НМ.

Алгоритмите за обучение може да имат различни реализации. При избор на стратегия на обучение се отчита класа решавани задачи, планираната точност и време за получаване на резултата.

Блокът за подбор на информативни признаци определя минимално необходимия или “разумен” брой връзки и неврони в мрежата. Най- важното следствие от работата на този модул е получаване на логически прозрачни мрежи, т.е. мрежи, чиито функциониране може да се опише и разбере с езика на логиката.

Качеството на обучение на НМ се определя от постигнатата грешка, която се изчислява в блока за оценка. Ако грешката превишава зададена стойност, се преминава на предходен етап от построяването на мрежовия модел и промяна на параметрите.

Базата от данни съдържа примерите за обучение на НМ. Те могат да съдържат числа, текст, графична информация.

Сигналите на входа на НМ трябва да се представят в съответния формат на данните. Те често се мащабират, за да се използва в максимална степен диапазона на изменението им. За всеки вид сигнал се прави определен вид предварителна обработка.

Интерпретаторът на отговорите взаимодейства с потребителя чрез интерфейса за извеждане.

За създаването на НМ в блока- конструктор се използват конструктивни и деструктивни методи.

Днес неврокомпютрите и технологиите на НМ са едни от най- бързо развиващите се направления в изкуствения интелект.

**15. Създаване на невронни мрежи с EasyNN.**

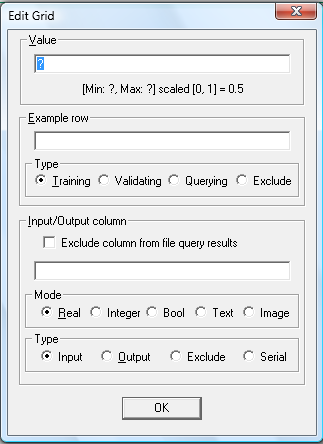
EasyNN-plus e продукт на фирма Neural Planner Software. Подходящ е за изграждане на малки невронни мрежи за експериментални задачи. Необходимо е предварително да бъде уточнена задачата, за която се изгражда мрежата: кои са основните величини, опредeлящи решението; как се ранжират по важност тези показатели; да се изберат примерите, в които се задават стойности за тези величини и съответния изход при всяка конкретна комбинация. Подборът на показателите е много съществен, защото така се избягват случаи, в които мрежата може да се обучи да взема решение върху неуместни характеристики. Понякога е необходимо установяването им чрез провеждане на разговори и анкети със специалисти. Върху въведената решетка от примери продуктът създава мрежа с параметри по подразбиране. Архитектурата на мрежата: като брой нива и брой на невроните във всяко от тях, може да бъде променена по- нататък. Използваната нелинейна функция в невроните е сигмоидната. За това изходът за всеки неврон се изчислява по формулата Out= 1/(1+ e (-net)), където е ≈ 2.7182, а net= ∑(xiwi). В мрежите се използва отклонение. Aлгоритъмът за обучение на мрежата е backpropagation. Изходната грешка за всеки неврон се изчислява като разлика между желания изход и получения изход (активацията) на неврона.

Например за отпускане на кредит на едно физическо лице е необходимо да се знае:

* нетна месечна заплата (Income);
* тип на трудовия договор (TypeContract) ;
* средно претеглено салдо по сметките (AverageAmount);
* професионален статус (ProfStatus);
* семейно положение (MaritalStatus);
* съдлъжник(Да/Не) (CoApplicant);
* брой деца (NbChildren);
* сума на кредита (LoanAmount);
* срок на кредита (в месеци) (LoanTerm);
* възможност за преференциален процент (TauxBon).

На база на конкретните стойности за тези величини се създава решетка от примери и се дава решението (Decision) – отпускане на кредит или отказ, което се взема в този случай и причината (Reason) за него.

Създаването на решетка от примери е по следната процедура. При вземане на нов файл с File->New се получава на екран празна решетка. Въвеждането на Enter извежда въпроса “Създаване на нов ред пример?”("Create new Example row?"), на който се отговаря с въвеждане на Enter за потвърждение. По същия начин се отговаря на въпроса “Създаване на нова входно/изходна колона?” (Create new Input/Output column?"). Новото въвеждане на Enter стартира диалоговия прозорец Edit Grid (фигура 22). Името на реда се въвежда в Example row и се посочва дали редът е за обучение-T, валидиране- V, заявка- Q или изключение- X. Името на колоната се въвежда в полето Input/Output column. Под това поле се избира типа на данните в колоната (Real, Integer, Bool, Text, Image) и типа на колоната. Самата стойност се въвежда в поле Value.

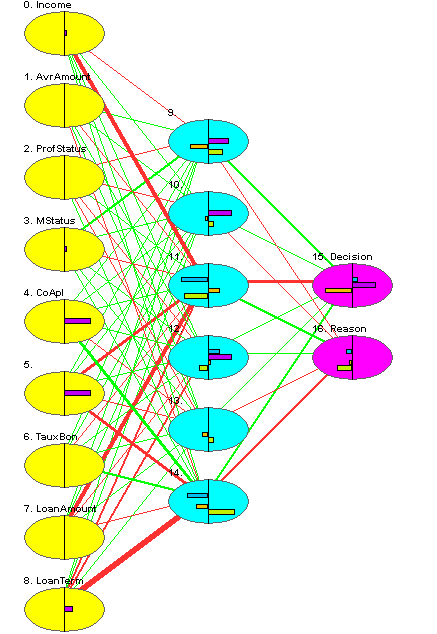


Фигура 22. Екран, чрез който се въвеждат примери в решетката на EasyNN

Всеки пример съдържа конкретни стойности за наблюдаваните величини и какъв изход се получава за тях. Липсващи или неизвестни данни в примерите се бележат с “?”. EasyNN позволява да се посочи каква стойност да се използва в този случай. Чрез команда Tools-> Options-> Missing data се задава да се вземе най- високата, най- ниската, средната стойност за колоната или тяхната медиана.

Всеки текстов стринг има като асоциирана стойност числова оценка, представляваща сума от произведенията на ASCII кода и позицията му ц стринга. Например Dog се асоциира с 529= 103\*1 + 111\* 2 + 68\*3.

При необходимост от редакция се използват командите от меню Еdit. За проверка дали примерите са достатъчни и дали са коректни се използва команда Action-> Check Gird.

Върху въведените примери EasyNN създава невронна мрежа (фигура 23) след команда Action-> New Network. Първоначално се препоръчва да се използват параметрите на мрежата по подразбиране. EasyNN може да създава невронни мрежи с до 3 скрити нива, но винаги се опитва най- напред вариант с едно, тъй като обикновено дава най- добри резултати. 

положителна връзка

Легенда: входен неврон отрицателна връзка изходен неврон

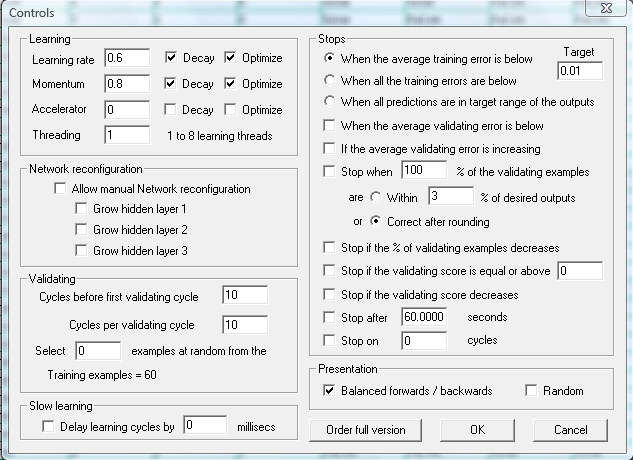
неврон от междинно ниво

Фигура 23. Невронна мрежа за разглежданата задача

Едно от често използваните правила е да се започне със скрито ниво неврони, съдържащи 30- 50% от броя неврони на входно ниво. Броят на входните неврони съответства на броя разглеждани величини, а на изходните- на броя изходни колони. В разглеждания пример изходите са два: решение за отпускане (отказ) на кредит и причина за решението.

В процеса на работа се създават няколко невронни мрежи като се прави избор между тях. Основният критерий за качество е постигане на най- ниско ниво на грешката за най- кратко време на обучение и получаване на верни отговори на запитванията към мрежата.

За създадената мрежа трябва да се зададат параметри на обучение (т.нар. контроли), да се посочи размера на целевата грешка и други параметри. За тази цел се използва команда Action-> Change Controls (фигура 24).



Фигура 24. Екран за установяване на контролни параметри на обучението

Параметърът Learning rate задава скорост на обучението. При създаване на мрежата тази стойност по подразбиране е 0.6. Може да приема стойности между 0.1 и 10. Малките стойности водят до бавно обучение, а тези над 1.5 водят до големи отклонения на грешката. Momentum- сила на обучението (по- малка сила води до повече цикли на обучение). Когато са избрани опциите Decay и Оptimize изборът на стойности за тези параметри е автоматичен. Изборът на Optimize настройва автоматично тази стойност, след изпълнение на няколко обучаващи цикъла с различни стойности. Decay автоматично намалява съответно скоростта и силата на обучение по време на обучение, ако грешката силно се колебае. По стойност е между 0 и 0.9.

В областта Stops се задават условия за край на обучението. Например:

* when аverage error is below- задава се целева грешка и когато тя стане по стойност под средната грешка - общата за примерите грешка, разделена на техния брой, обучението спира. На фиг.24 е зададено условието средната грешка да падне под 0.01.
* when аll the training errors are below- когато целевата грешка стане по- малка от грешките на обучение на всеки пример и т.н.

Обучението започва след избор на команда Action-> Start Learning. Продуктът дава възможност графично да се следи промяната на грешката при избор на команда View-> Graph.

Една възможност за спиране на обучението е, когато средната грешка (Average error) падне под нивото на целевата грешка (Target error).

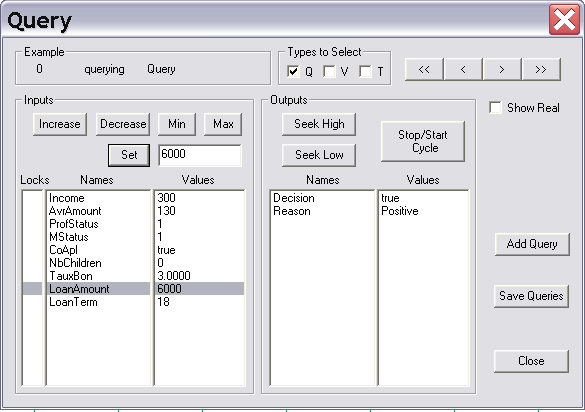
Обучената мрежа повече не променя теглата и може да се използва за отговор на запитвания към нея. Използва се команда Query -> Query (фигура 25). Нова заявка се прави с бутон Add Query. Задават се конкретни стойности за всяка величина, които могат да са различни от тези в решетката от примери. Това става като се избере в левия панел дадена величина и с бутоните:

* Мin (Max) се избира нейната минимална (максимална) стойност от въведените примери. Стойността е преобразуваната входна.
* Increase (Decrease) се увеличава (намалява) стойността. За цели числа увеличението (намалението) е с 10%, а за реални- с 1%.

Неизвестните величини имат за вход “?”.

В полето към бутона Set се въвежда конкретна стойност за величината, която се копира в левия панел след натискане на бутона.

В десния панел се вижда решението на мрежата.



Фигура 25. Задаване на заявка към примерната НМ

Бутоните Seek High (Seek Low) се използват за търсене на най- високата (ниската) стойност за избрана величина, която резултира в най- високата (ниската) стойност на изхода. Търсенето се стартира (спира) чрез бутон Stop/Start Cycle.

В някои случаи при заявка намирането на най- близката асоциирана стойност до зададената в Output Column се задава със серия вълнички „~” преди отговора. Една вълничка означава отговор най- близък по стойност, две- следващ и т.н. Това е свързано с асоциирането на стойност за всеки текстов стринг. Например, ако се получи като отговор стойност 712, за която няма асоцииран текст. EasyNN намира най- близката асоциирана стойност в изходната колона (например 713- cat) и поставя пред нея вълничка в знак на степен на близост (~ cat).

Някои невронни мрежи могат да се обучат и използват с по- малко връзки между нивата неврони от автоматично създадените. Чрез команда Action->Reduce Connections и въвеждане на новия брой връзки се правят промени на броя им. Трябва да се има превид, че всеки неврон на всяко ниво може да има най- малко една връзка.

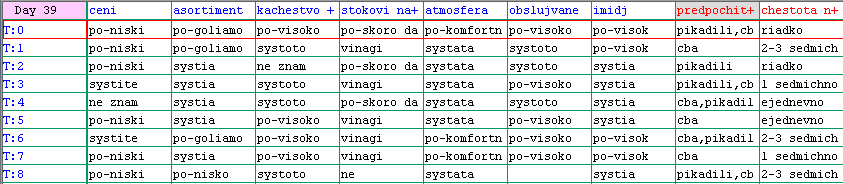
Практически пример:

Чрез невронна мрежа да се оцени как влияят върху **предпочитанията към търговски обект** и **честотата на пазаруване** във верига „СВА” и верига „Пикадили” в гр.Варна, следните седем основни фактора на потребителския избор: цени, разнообразие и наличност на асортимента, качество на предлаганите стоки, атмосфера, качество на прякото обслужване и имидж на обекта.

Приложена е скала за оценка на детерминантите на потребителския избор, например „по- високо”, „същото” и „по- ниско” равнище в сравнение с конкурентните търговски обекти, както и „не знам”, „винаги” и др.п.

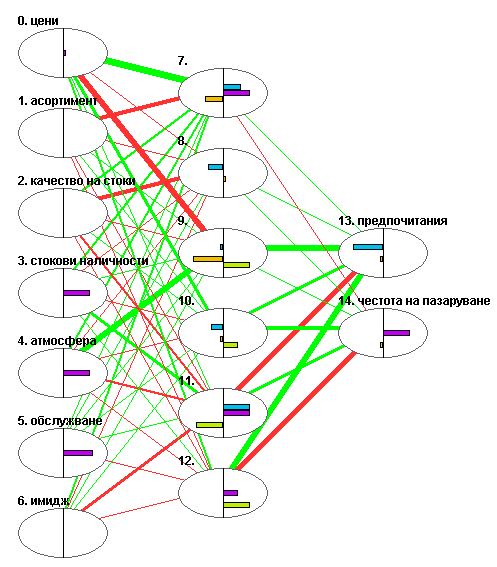
Решението се предлага на база на софтуерния продукт EasyNN, който работи с многослойни перцептрони.

На Фигура 26 е представена част от обучаващото множество, използвана в експеримента[[2]](#footnote-2). Като изходни величини се задават **предпочитанията към търговски обект** и **честотата на пазаруване,** а като входни- седемте фактора на потребителския избор.



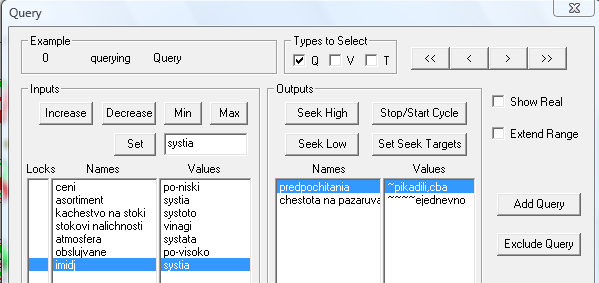
Фигура 26. Част от обучаващото множество, използвано в експеримента

Архитектурата на мрежата, използвана в експеримента е показана на **Фигура** 27. Броят на входните неврони съответства на броя величини, за които се въвежда информация. Броят на изходните неврони представлява броя на наблюдаваните изходни показатели.



Фигура 27. Невронна мрежа за експеримента

При зададена заявка (запитване)- **Фигура 28** към мрежата с конкретни стойности за входните величини, например цени - “по- ниски”, асортимент - “същия” (както на конкурентните обекти), качество на стоките- “същото”, стокови наличности - “винаги” (клиентът намира търсената стока), атмосфера – “същата”, ниво на обслужване - “по- високо” и имидж - “същия”, мрежата извежда за предпочитания на клиента - “пикадили, cba” и честота на пазаруване - “eжедневно”.



Фигура 28. Заявка към невронната мрежа с конкретни стойности за входните величини

Мениджърът може да извършва симулации чрез заявки към мрежата, за да оцени как биха повлияли на потребителските предпочитания инвестициите в по- добро качество на стоките, по- богат асортимент, по- комфортна атмосфера и др.п. Много важно е също, че може да се проследи степента на влияние на входните величини върху изходния резултат (фиг. 29). Както се вижда величината с най- голямо влияние върху потребителските предпочитания са цените на стоките, след това атмосферата в търговския обект и т.н.

# 

Фигура 29. Влияние на входните величини върху изходния резултат

# Получените коректни резултати от прилагането на невронните мрежи за конкретна практическа задача, дават основание те да бъдат прилагани, когато могат да се съберат достатъчно данни по проблема.

1. Например централен процесoр с 108 логически елемента, с оперативна памет 1010 бита, с капацитет на диска 1011 бита, продължителност на цикъла на обработка 10-9 секунди, количество обновявания на паметта 109 бита/ секунда. [↑](#footnote-ref-1)
2. Апробацията на модела е извършена на основата на анкетно проучване на потребителски оценки. [↑](#footnote-ref-2)