

ВИВЧЕННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ У КУРСІ "ОСНОВИ ШТУЧНОГО ІНТЕЛЕКТУ"

У статті розглядаються питання вивчення штучних нейронних мереж як однієї з моделей представлення знань.

Ключові слова: *штучні нейронні мережі, перцептрон, порогова функція активації.*

Останнім часом можна спостерігати зростання інтересу до так званих штучних нейронних мереж, тому питання щодо їх вивчення повинно обов'язково бути включеним у курс "Основи штучного інтелекту".

За [1] штучні нейронні мережі (ШНМ) – математичні моделі, а також їх програмна та апаратна реалізація, побудовані за принципом функціонування біологічних нейронних мереж – мереж нервових клітин живого організму. Архітектура і принцип дії штучних нейронних мереж базується на аналогії з мозком живих істот.

Штучні нейронні мережі виникли в рамках досліджень з питань штучного інтелекту. Історія ШНМ починається з 1943 року, коли У.Маккалок та У.Пітс запропонували модель біологічного нейрона, сформулювавши основні положення щодо роботи людського мозку.

Загальновідомо, що людський мозок має значні переваги у порівнянні з комп'ютером у багатьох задачах, наприклад у задачах розпізнавання образів.

Виділяють кілька принципових відмінностей людського мозку щодо опрацювання даних у порівнянні зі звичайним комп'ютером [2]:

1. Здатність до навчання на прикладах.
2. Здатність до узагальнень. На прикладах конкретних екземплярів якогось об'єкта, наприклад фото різних літаків, у людському мозку формується узагальнений образ цього об'єкта, тобто літака. Далі спостерігаючи будь-який об'єкт, людина може визначити, літак це, чи ні.
3. Паралельність опрацювання даних (наприклад зображення людиною аналізується не за пікселями, а відразу повністю).
4. Висока надійність мозку. З віком люди втрачають значну частину нервових клітин, проте залишаються у здоровому глузді і міцній пам'яті.
5. Асоціативність людської пам'яті, тобто здатність знаходити потрібні дані за малою їх частиною.

Всі ці переваги спонукали до досліджень, які б надавали можливість з'ясувати, як працює людський мозок, і спроб змоделювати таку роботу.

Було з'ясовано, що мозок складається з нейронів. В тіло біологічного нейрона вхідні сигнали поступають через синапси у вигляді електричних імпульсів, які там накопичуються. Коли рівень електричного заряду у біологічному нейроні досягне певної величини, цей заряд передається у вихід біологічного нейрона, який називають аксоном.

У.Маккалок та У.Пітс запропонували модель біологічного нейрона, як перемикача, що отримує дані від інших нейронів. Отримані дані помножуються на значення, співставлені з синапсами, і підсумовуються. Якщо отримане значення перевищує задану межу, нейрон активізується, в протилежному випадку він залишається неактивним. У 60-х роках 20-го століття було доведено, що за такою моделлю можна виконувати складні операції розпізнавання образів.

Сформулюємо більш строго поняття штучного нейрона. Штучним нейроном називається математичний об'єкт, що має n входів і 1 вихід, через які передаються числові дані.

Входи пов'язують штучний нейрон з іншими штучними нейронами або зовнішнім джерелом даних і називаються зв'язками. Вхідні дані позначатимемо x_i . З кожним зв'язком пов'язане число, що називається ваговим коефіцієнтом зв'язка. Позначимо ці числа w_i . Опрацювання даних відбувається за два етапи.

На першому етапі обчислюється зважена сума $S = w_0x_0 + \dots + w_nx_n$.

На другому етапі обчислюється значення y як деяка функція від зваженої суми: $y = f(S)$; y називають функцією активації.

Функції активації бувають різних видів. Історично першою була східчаста функція $f(S)$, за якою повертається нуль, коли $S < 0$, і 1 у протилежному випадку. Тобто за аналогією з біологічним нейроном – коли загальний імпульс перевищить критичне значення, генерується імпульс зі значенням 1, в протилежному випадку нейрон залишається у спокої.

Існують і інші функції активації, наприклад логістична функція (сигмоїд):

$$f(S) = \frac{1}{1 + e^{-aS}}$$

При зменшенні α графік сигмоїда стає більш пологим, при збільшенні α графік сигмоїда за зовнішнім виглядом стає схожим на графік східчастої функції.

З таких штучних нейронів складають штучну нейронну мережу. Штучні нейронні мережі можуть мати різні структури. Однією з найбільш поширених структур є багатошарова. Шар – це сукупність штучних нейронів, які не з'єднані між собою, але з'єднані з штучними нейронами попереднього та наступного шарів. Зовнішні вхідні сигнали подають на входи нульового шару мережі, а виходами мережі є вихідні сигнали останнього шару. Багатошарові ШНМ називають перцептронами.

Серед всіх багатошарових ШНМ виділяють:

1. Мережі прямого розповсюдження. В цих мережах передають сигнали тільки від попереднього шару до наступного. Як правило, кожен вихідний сигнал n -го шару передається на вхід всіх нейронів $(n+1)$ шару.

2. Мережі із зворотними зв'язками. В таких мережах дані також передаються з наступних шарів до попередніх.

Найпростішою ШНМ є мережа, що складається з одного штучного нейрона з n входами, одним виходом і пороговою функцією активації (елементарний перцептрон). На входи подаються значення x_i . Для кожного з входів визначено ваговий коефіцієнт w_i , а також визначено поріг. На виході такої ШНМ буде 1, коли сума $S = w_0x_0 + \dots + w_nx_n$ перевищить поріг і -1 в протилежному випадку. Таку мережу можна використовувати для розв'язування задачі класифікації двох класів: якщо на виході 1, то поданий на вхід вектор належить до першого класу, в протилежному випадку – до другого.

Збільшуючи кількість нейронів у шарі, та кількість шарів, можна розв'язувати, наприклад, досить складні задачі розпізнавання образів.

Розглянемо тепер ШНМ, в якій функція активації є неперервною. В цьому випадку виходом нейронної мережі є неперервна функція її входів. Таким чином таку ШНМ можна використовувати для апроксимації функцій. За узагальненою теоремою Стоуна за допомогою ШНМ з довільною функцією активації можна апроксимувати довільну неперервну функцію.

Для того, щоб за допомогою ШНМ розв'язувати задачі, наприклад розглянуті вище задачі розпізнавання образів або апроксимації, потрібно правильно дібрати вагові коефіцієнти нейронів (w_i). Для цього ШНМ потрібно "навчити". Існують алгоритми "навчання" 3-х видів.

1. Алгоритми навчання з учителем. За цим алгоритмом за допомогою ШНМ розв'язують ряд прикладів, кожен з яких складається з двох частин:

– значення, що подають на вхід;

– значення, яке повинно бути на виході.

В процесі "навчання" вагові коефіцієнти модифікують таким чином, щоб за вхідними даними отримувати вихідні значення, максимально наближені до бажаних.

2. Алгоритми "навчання з заохоченням". В прикладах, що пропонуються розв'язати за допомогою ШНМ, не вказують бажане вихідне значення, але після проходження кожного прикладу виставляється оцінка, як було виконане завдання, добре чи погано.

3. Алгоритми "навчання без вчителя". За допомогою ШНМ розв'язують набір прикладів (без бажаного значення на виході) і в процесі їх опрацювання в ШНМ відбуваються певні процеси самоорганізації, що призводять до модифікації вагових коефіцієнтів так, що за допомогою ШНМ стає можливим розв'язати певну задачу.

Одним з найбільш популярних алгоритмів "навчання з учителем" є алгоритм зворотного розповсюдження. За цим алгоритмом спочатку вагові коефіцієнти вибирають випадковим чином. Є певний набір "навчальних" прикладів, що складається з пар X, D (вектор X на вході та бажане значення D на виході). Через Y позначено реальний вихід мережі. "Навчання" полягає в тому, щоб дібрати вагові коефіцієнти W так, щоб мінімізувати певну цільову функцію, в якості якої розглядають суму квадратів помилок ШНМ, тобто різниць між Y та D . Мінімізувавши цю функцію, отримують розв'язок за методом найменших квадратів.

Для елементарного перцептрона можна запропонувати такий алгоритм навчання з учителем:

1. Вибирають вагові коефіцієнти w_i випадковим чином.

2. На вхід подають значення першої частини x_i поточного тренувального прикладу і обчислюють y .

3. Порівнюють вихід (y) з бажаним значенням (d). Якщо отримане значення співпадає з бажаним, то вибирають наступний навчаючий приклад і переходять по пункту 2, в протилежному випадку переходять до пункту 4.

4. Обчислюють нові значення вагових коефіцієнтів:

(нові w_i) = (попередні w_i) + $d * x_i$ і переходять до пункту 2, вибравши наступний навчаючий приклад.

Використання ШНМ дозволяє розв'язувати задачі з багатьох предметних галузей, наприклад розпізнавання образів, прогнозування на фондовому ринку, визначення ризиків надання кредиту, управління роботом та інші.

Для формування у студентів компетентностей щодо створення найпростішої штучної нейронної мережі (елементарного перцептрона), доцільно розглянути з ними наступний приклад.

Приклад. Напишемо програму мовою Object Pascal, за якою можна проводити розпізнавання образів, а саме відрізати прямокутники від відрізків.

Прямокутники і відрізки будуть задаватися зафарбованими точками на рисунку розміром $\delta x \delta$. Для задання такого рисунка знадобиться масив з 64 цілих чисел. Одиниці в цьому масиві визначатимуть зафарбовані точки, нулі – не зафарбовані.

За програмою спочатку відбувається "навчання" ШНМ шляхом виклику процедури *Teach* з параметрами, що відповідають описаним в програмі прямокутникам (*xk1, xk2, xk3, xk4, xk5, xk6*) та відрізкам (*xp1, xp2, xp3, xp4, xp5, xp6*), а потім класифікація рисунків *xz1, xz2, xz3, xz4*, відмінних від тих, що використовувались у "тренуванні".

```

uses SysUtils;
const n=64;
type tvector=array[1..n] of integer;
const xk1:tvector=(1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1);
{Опис констант xk2, xk3, xk4, xk5, xk6 }
...
const xp1:tvector=(1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
{Опис констант xp2, xp3, xp4, xp5, xp6}
...
const xz1:tvector=(0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0,
0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1);
{Опис констант xz2, xz3, xz4}
...
type tperceptron=record
    w:array[1..n] of integer;
    porig:integer;
end;
var perceptron:tperceptron;
function ask(x:tvector):integer;
var i,s:integer;
begin
    s:=0;
    for i:=1 to n do
        s:=s+x[i]*perceptron.w[i];
    if (s>perceptron.porig) then
        ask:=1
    else
        ask:=-1;
    end;
end;
procedure init_perceptron;
var i:integer;
begin
    randomize;
    for i:=1 to n do
        perceptron.w[i]:=random(10);
    perceptron.porig:=70;
end;

```

```

end;
procedure teach(x:tvector;d:integer);
var i:integer;
begin
  if (d<>ask(x)) then
    for i:=1 to n do
      perceptron.w[i]:=perceptron.w[i]+d*x[i];
    end;
end;
begin
  init_perceptron;
  teach(xk1,1); teach(xk2,1); teach(xk3,1); teach(xk4,1);
  teach(xk5,1);teach(xk6,1); teach(xp1,-1); teach(xp2,-1);
  teach(xp3,-1); teach(xp4,-1); teach(xp5,-1); teach(xp6,-1);
  if (ask(xz1)=1) then writeln('Rectangle') else writeln('Line');
  if (ask(xz2)=1) then writeln('Rectangle') else writeln('Line');
  if (ask(xz3)=1) then writeln('Rectangle') else writeln('Line');
  if (ask(xz4)=1) then writeln('Rectangle') else writeln('Line');
end.

```

Використані джерела

1. Штучна нейронна мережа : матеріал з Вікіпедії – вільної енциклопедії. [Електронний ресурс] – Режим доступу: http://uk.wikipedia.org/wiki/Штучна_нейронна_мережа – 5.09.2012 р.
2. Нейронные сети. [Електронний ресурс] – Режим доступу: www.inf-man.ru/download/sysan/lecture02sa.doc – 5.09.2012 р.

Horoshko Y.V.

STUDY OF ARTIFICIAL NEURON NETWORKS IN THE COURSE "BASIS OF ARTIFICIAL INTELLIGENCE"

In the article the questions of study of artificial neuron networks are examined as one of models of representation of knowledges.

Keywords: *artificial neuron networks, perceptron, threshold function of activating.*

Стаття надійшла до редакції 10.08.2013 р.

