

Тема 1/Занятие 8/Лекция

Обучение на невронни мрежи.

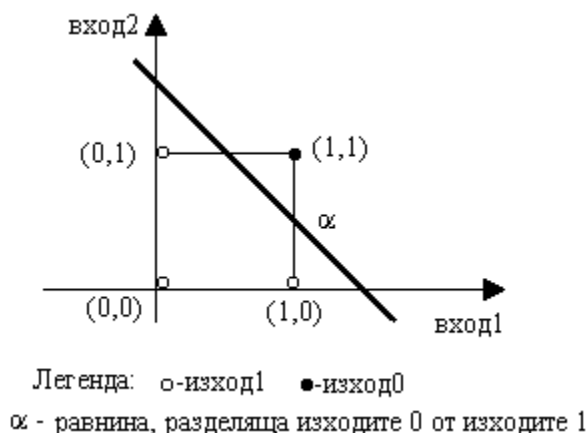
1. Съдържание на процеса на обучение на изкуствена невронна мрежа.

Обучението на невронната мрежа е процес, при който тя се настройва да реагира по определен начин на определени входни въздействия. По този начин се постига получаване на достоверни резултати от работата ѝ.

Най-общо обучението на невронната мрежа се изразява в промяна на нейните компоненти, така че тя да реагира по желания начин на определени входни въздействия. Теоретично една невронна мрежа може да се учи чрез:

- създаване на нови връзки между невроните;
- изтриване на съществуващи връзки;
- промяна на тегловите коефициенти на връзките;
- промяна на праговите стойности на невроните;
- промяна на една или повече от следните три функции (активиращата, функцията на разпространение и изходната функция),
- създаване на нови или изтриване на съществуващи неврони (и на връзките помежду им).

Не всички задачи са решими чрез обучението на една невронна мрежа. Подходящи са тези от тях, които са **линейно отделими**. Това означава, че за задачата може да се намери равнина в пространството, която да разделя множествата от различни изходни резултати, получени при отделните комбинации входни сигнали.



Фиг. 4. Линейна отделимост на логическата функция конюнкция

Ако такава равнина не може да бъде построена, задачата не може да бъде решена чрез невронна мрежа. Пример за линеино отделима задача е логическата функция конюнкция (фиг.4). Входните сигнали и резултатът, които трябва да се получат са посочени в таблица.1.

таблица.1. Таблица на истинност на логическа функция конюнкция

Вход 1	Вход 2	изход
0	0	0
0	1	0
1	0	0
1	1	1

От фигурата се вижда, че може да се построи равнина, отделяща изход 1 от изходите 0. Такава задача може да бъде решена от обикновен неврон.

Способността за обучението е основна особеност на невронните мрежи. То се реализира по подходящи алгоритми чрез реализация на многократна промяна на тегловните коефициенти на връзките между невроните.

Комбинацията от входни сигнали за които се търси правилната реакция от мрежата се нарича **обучаващо множество**.

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.

Обучаващия процес започва с въвеждане на първия пример на входното ниво на мрежата. Произвежда се изход, който обикновено е далеч от целта, защото е базиран на текущото състояние на връзките. Поради произволно избраните първоначално тегла, резултатът се различава от желания. Сравнява се получения и желания изход. В зависимост от размера на грешката теглата на всички връзки се настройват, така че отклонението да се намали. За тази цел се използват подходящи алгоритми (обучаващи правила). Процесът се повтаря със следващия пример и продължава докато се минимизира грешката или се постигне поставената цел. Грешката се намалява чрез постепенна настройка на теглата в невронната мрежа. Когато обучаващия процес се преустанови, мрежата е готова за използване. Постига се стабилно състояние. Обикновено теглата не се променят за времето за което мрежата се използва по задачата, за която е обучена.

От съществено значение при обучението на изкуствени невронни мрежи е да се определи кога да се спре процеса на обучение. Тъй като способността да се правят обобщения е от фундаментално значение за способността на мрежите, прекомерното обучение, прекомерно нагаждане или прекаляването с процеса на трениране е сериозен проблем.

Друга важна особеност в определянето и осигуряването на необходимото количество данните за обучение. Необходимо е да се използва максимално голяма база данни, с която системата може да бъде обучена. Добре е да бъдат използвани достатъчно голям набор от данни, поради факта че някои шаблони на поведение не могат да бъдат открити в малки бази данни.

2. Обучение с учител

Контролираното обучение изисква много шаблони, които да служат като образци. Всеки шаблон принадлежащ от комплекта за обучение съдържа входни стойности и съответните целеви изходни. Мрежата изчислява целевата стойност на базата на зададените входни данни чрез минимизиране на грешката на изхода си до достигане на целевия изход. Това се реализира посредством непрекъснато коригиране на теглата на връзките си чрез един повтарящ се процес на обучение, който на практика представлява трениране.

Най-често срещаният алгоритъм за обучение на мрежата е **алгоритъм с обратно разпространение на грешката**.

Обратното разпространение на грешката по време на обучението на изкуствена невронна мрежа е процеса по разпространение на грешка в системата в посока, обратна на придвижването на входящата информация, т.е. от изходния към входния слой на мрежата. Изходящия слой на мрежата е единствения, който има целева стойност, с която да се сравнява получения резултат и докато грешката се разпространява по обратният път към входящия слой чрез възлите, тежестите на връзките между отделните неврони се променят. Обучението продължава докато грешката в теглата е достатъчно незначителна, за да бъде приета. Видът на активиращата функция, използвана

във възлите на невронната мрежа може да окаже влияние при определяне на вида на данните, с които се обучава мрежата.

3. Обучение без учител

Използват се и **самообучаващи се невронни мрежи**. Този тип мрежи сами откриват най-характерните черти на категориите, които трябва да различават и изработват собствено представяне на входните сигнали.

и не се нуждае от конкретни стойности на изхода за провеждане на обучението. За всеки един от шаблоните подавани на входния слой на мрежата се допуска, че принадлежи към отделен клас параметри. По този начин, процесът на обучение се свежда до това да се остави мрежата да разкрие тези класове. Този метод не е толкова популярен колкото контролираното обучение и не е подходящ да се използва в този труд. Поради тази причина няма да бъде разглеждан по-нататък.

Два популярни метода за самообучение на невронни мрежи са: **обучение, основано на главните компоненти на данните и състезателно обучение.**

Усъвършенстващо обучение

Усъвършенстващото обучение е хибриден метод за обучение. При него на мрежата не се задават целеви резултати, но в същото време мрежата се информира, ако изчислените изходни данни се движат в правилната посока или не.

4. Алгоритъм за обучение на еднослоен перцептрон

Перцептронът трябва да реши проблема с класификацията чрез двоични входни сигнали. Наборът от входни сигнали ще бъде означен с n -мерния вектор x . Всички елементи на вектора са булеви променливи (променливи, които приемат стойностите "True" или "False"). Понякога обаче е полезно да се оперира с числени стойности. Ще приемем, че стойността "false" съответства на числовата стойност 0, а стойността "True" съответства на 1.

Перцептронът е устройство, което изчислява следната система от функции:

$$\psi = \left[\sum_{i=1}^m w_i x_i > \theta \right], \quad (1)$$

където w_i са теглата на перцептрона,

θ е прагът, x_i са стойностите на входните сигнали, скобите $[]$ означават прехода от булеви (логически) стойности към числени стойности според правилата, посочени по-горе.

Обучението на Perceptron се състои в коригиране на коефициентите на тегло. Нека има набор от двойки вектори (x^α, y^α) , $\alpha = 1, \dots, p$, наречен набор за обучение.

Невронната мрежа ще бъде обучена на даден тренировъчен набор, ако, когато всеки вектор, който се подава към мрежовите входове, съответният изходен вектор се получава всеки път на изходите.

Методът на обучение, предложен от Ф. Розенблат, се състои в итеративно коригиране на матрицата на теглото, което последователно намалява грешката в изходните вектори. Алгоритъмът включва няколко стъпки:

Стъпка 0 Първоначалните стойности на теглата на всички неврони $W(t=0)$ се приемат за случайни

Стъпка 1 Входното изображение се представя на мрежата, в резултат на което се формира изходното изображение.

Стъпка 2 Изчислява се векторът на грешката $\delta^\alpha = (y^\alpha - \tilde{y}^\alpha)$ направена от мрежата на изхода. Другата идея е, че промяната на тегловния вектор в областта на малките грешки трябва да бъде пропорционална на изходната грешка и равна на нула, ако грешката е нула.

Стъпка 3 Тегловният вектор се модифицира по следната формула: $W(t + \Delta T) = W(t) + \eta x^\alpha \cdot (\delta^\alpha)^T$, където $0 < \eta < 1$ е скоростта на обучение.

Стъпка 4 Стъпки 1-3 се повтарят за всички вектори за обучение. Един цикъл на последователно представяне на цялата проба се нарича епоха. Обучението завършва след няколко епохи: а) когато итерациите се сближат, т.е. тегловният вектор спира да се променя, или б) когато общата абсолютна грешка, сумирана за всички вектори, стане по-малка от някаква малка стойност.

Посочените в алгоритъма стъпки могат да бъдат интерпретирани по следния начин:

подаваме на входа на персептрона вектор x , за който верният отговор вече е известен. Ако изходът на персептрона съвпада с правилния отговор, тогава не е необходимо действие. В случай на грешка е необходимо да обучите персептрона да реши правилно този пример. Грешките могат да бъдат два вида.

Първият тип грешка: изходът на персептрона е 0, а верният отговор е 1. За да може персептронът да даде правилен отговор, е необходимо сумата от дясната страна на (1) да стане по-голяма. Тъй като променливите приемат стойности 0 или 1, може да се постигне увеличение на сумата чрез увеличаване на теглата w_i . Въпреки това, няма смисъл да се увеличават теглата, когато променливите x_i са равни на нула. Следователно теглата w_i трябва да бъдат увеличени за тези променливи x_i , които са равни на 1.

Първо правило. Ако изходът на персептрона е 0, а правилният отговор е 1, тогава е необходимо да се увеличат теглата на връзките между едновременно активни неврони. В този случай изходният персептрон се счита за активен.

Вторият тип грешка: изходът на персептрона е 1, а верният отговор е нула. За да научите правилното решение на този пример, сумата от дясната страна на (1) трябва да бъде намалена. Следователно е необходимо да се намалят теглата на връзките w_i за тези променливи, които са равни на 1 (тъй като няма смисъл да се намаляват теглата на връзките за променливи x_i , равни на нула). Също така е необходимо тази процедура да се извърши за всички активни неврони от предишните слоеве. В резултат на това получаваме второто правило.

Второ правило. Ако изходът на перцептрона е единица, а правилният отговор е нула, тогава е необходимо да се намалят теглата на връзките между едновременно активни неврони.

По този начин процедурата за обучение се свежда до последователно изброяване на всички примери от набора за обучение, като се използват правилата за обучение за погрешно решени примери. Ако след следващия цикъл на представяне на всички примери се окаже, че всички са решени правилно, тогава процедурата за обучение е завършена.

Теорема за конвергенцията на персептрон. Ако има параметър вектор w , за който персептронът решава правилно всички примери от набора за обучение, тогава, когато персептронът се обучава съгласно описания по-горе алгоритъм, решението ще бъде намерено в краен брой стъпки.

Теорема за зацикляне на персептрон. Ако няма параметърен вектор w , за който персептронът решава правилно всички примери от обучителната извадка, тогава, когато персептронът се обучава съгласно това правило, векторът на теглото ще започне да се повтаря след краен брой стъпки.

По този начин тези теореми гласят, че след стартиране на процедурата за обучение на персептрон, след крайно време получаваме или обучен персептрон, или отговорът, че този персептрон не може да научи задачата.

Литература:

1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на http://www.dkriesel.com/en/science/neural_networks, посетено на 12.08.2022 г.

2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5-03-002115-9.
4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0-13-273350-1.
5. Гульнара Яхьяева, Лекция 3. Персептроны. Обучение персептрон, доступно на https://intuit.ru/studies/courses/88/88/print_lecture/20531