



»Лекционен курс

»Изкуствен интелект

Математически основи на машинното учене



Увод

- » В основата на статистическото машинно учене лежи значителен по обем и сложност математически апарат, който не може да бъде обхванат в рамките на един лекционен курс.
- » Тук ще демонстрираме само използването на избрани математически концепции и модели.



Увод

- » Много интелигентни задачи, изпълнявани от хора, могат да бъдат изразени чрез математически функции, често наричани **целеви функции**.
- » Машините, изпълняващи интелигентни задачи, трябва да са в състояние да моделират целеви функции.
- » Търсенето на автоматизирани решения среща две сериозни трудности:
 - > Освен че са произволно сложни, целевите функции, лежащи в основата на различни проблеми от реалния живот, са напълно различни една от друга - едва ли има някакъв общ модел.
 - > За повечето проблеми не знаем основната целева функция.



Увод

- » Въпреки това се търси механизирано повторяемо решение за изпълнение на реални интелигентни задачи, като не искаме да започваме от нулата и да проектираме функция за всеки такъв проблем.
- » Тук могат да помогнат **невронните мрежи**, като осигурят унифицирана рамка за моделиране на изключително голямо разнообразие от произволно сложни функции.



Увод

- » Докато цялостната невронна мрежа **моделира сложна функция**, нейният градивен елемент **невронът** представлява **сравнително проста функция**.
- » Пълната невронна мрежа се състои от **много неврони с претеглени връзки** между тях.
- » Тя може да бъде изградена така, че да **апроксимира всяка произволна целева функция** чрез **манипулиране на броя на невроните, връзките между тях и теглата на връзките**.



Увод

- » Разнообразието и сложността на функциите, които една невронна мрежа може да представи, са известни като нейната **експресивна сила**.
- » Експресивната сила нараства с броя на невроните в невронната мрежа и броя на връзките между тях.



Аспекти

- » Колкото по-сложна е целевата функция, толкова по-изразителна сила ще е необходима в невронната мрежа, която я моделира.
- » За да направим модел на невронна мрежа да съответства на конкретен интересуващ ни проблем можем да коригираме следните два аспекта на невронната мрежа:
 - > **Архитектура** - броят на невроните и връзките между тях.
 - > **Стойности на параметрите** - теглата на връзката между невроните.



Архитектура

- » Архитектурата обикновено се избира въз основа на естеството на проблема.
- » Някои популярни архитектури се използват многократно и инженерът на невронни мрежи обикновено избира архитектура, за която исторически е известно, че е ефективна за проблем, подобен на решавания.
- » След като архитектурата е зададена трябва да определим стойностите на параметрите чрез процес, наречен **обучение**.



Видове невронни мрежи

- » Невронните мрежи могат да бъдат класифицирани в два основни класа:
 - > Контролирани.
 - > Неконтролирани.
- » В контролираните невронни мрежи ние идентифицираме желаните изходни стойности, съответстващи на набор от извадени входни стойности за проблема, който се опитваме да разрешим.
- » Желаният изход за тези извадкови входове обикновено се избира ръчно с помощта на процес, наречен **етикетиране**.
- » Цялостният набор от двойки <извадка на вход, желан изход> представлява данните за контролирано обучение.



Обучение

- » По време на обучение стойностите на параметрите (известни още като тегла) се коригират така, че изходните данни на мрежата при тренировъчни входи да съответстват на съответната реална истина възможно най-близо.
- » Ако всичко върви добре, в края на обучението, имаме невронна мрежа, чиито изходи са близки до реалната истина.
- » Тази обучена невронна мрежа след това се разгръща в реалния свят, където извършва изводи - генерира изходни данни на входи, които никога преди не е виждала.



Обучение

- » Ако сме избрали архитектура с достатъчно изразителна мощност и правилно сме обучили мрежата с адекватни данни за обучение, тя трябва да излъчва точни резултати по време на извода.
- » Обикновено не можем да гарантираме правилни резултати по време на извода, а можем само да направим вероятностно твърдение, че нашият резултат има p вероятност да бъде правилен.



Етикиране

- » Неконтролираните невронни мрежи не се нуждаят от ръчно етикетирана реална истина - те просто работят върху входящите данни за обучение.
- » Ръчният труд, свързан с етиктирането на данните за обучение, е скъп и досаден.
- » Следователно, значителни изследователски усилия се насочват към невронни мрежи, които са неконтролирани, полуконтролирани (част от данните за обучение се етиктират ръчно) или самоконтролирани (маркираните данни за обучение се създават програмно, а не ръчно).
- » В момента неконтролируемата и полуконтролираната невронна мрежова технология е по-малко развита и е по-трудно да се постигнат желаните нива на точност с тях.



- » Важно е да се отбележи, че никъде в процеса на избор на архитектура или обучение не се нуждаем от представяне в затворена форма или на апроксимираната функция (целевата функция) или на апроксиматорната функция (функцията за моделиране).
- » В повечето случаи е невъзможно да се знае целевата функция - всичко, което знаем, са двойки примерен вход и реална истина (данни за обучение).
- » Що се отнася до функцията за моделиране, дори когато знаем архитектурата на моделиращата невронна мрежа, цялостната функция, която тя представлява, е толкова сложна, че е практически неразрешима.
- » По този начин фактът, че не е необходимо да знаем целта или функцията за моделиране в затворена форма, е това, което прави технологията практична.



Изразяване на проблеми от реалния свят: целеви функции

- » Много проблеми от реалния свят могат да бъдат изразени като целеви функции, като всички имащи отношение към резултата количествено измерими променливи представляват **входните променливи** (обикновено означени с x или последователност от входни променливи често се изразява като вектор x).
- » Входните променливи се изразяват като числови единици, като например скалари, вектори, матрици и т.н.
- » Резултатите също се изразяват като числови променливи, наречени **изходни променливи** (обикновено се означават с y).



Изразяване на проблеми от реалния свят: целеви функции

- » Като се има предвид конкретен вход моделиращата функция доставя изход, който е решение на проблема за този конкретен вход.
- » Общата целева функция обикновено се изразява като $y = f(x)$.
- » Често се използват индекси за обозначаване различни елементи на вектор (x_0, x_1, \dots, x_i) и горни индекси за обозначаване входни случаи, както $y^{(0)} = f(x^{(0)})$, $y^{(1)} = f(x^{(1)})$, \dots , $y^{(i)} = f(x^{(i)})$.



Променливи

- » Числените количества могат да се появят в две различни форми:
 - > Непрекъснатата.
 - > Дискретна.
- » Непрекъснатите променливи могат да приемат всяка от безкрайно много стойности на реални числа в даден диапазон.
- » Дискретните променливи могат да приемат една от краен набор от разрешени стойности, където стойността представлява някаква категория.
- » Специален случай е двоичната променлива, където се различават само две категории.



Логически функции в проблеми от реалния свят

- » Това са функции, чиито входове и изходи са двоични променливи: променливи, които могат да приемат само две стойности:
 - > 0 - известен още като „не“ или „не задействай“.
 - > 1 - известен още като „да“ или „задействан“.
- » Машините, емулиращи логически функции, често се добавят към отделни машини, изпълняващи други задачи.

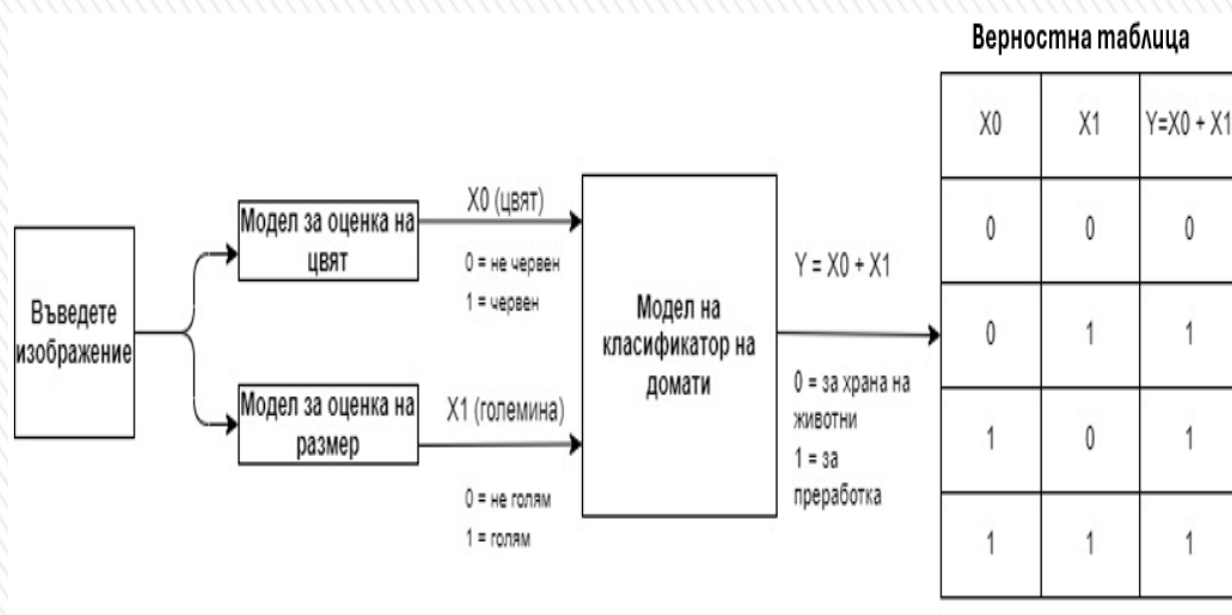


Пример

- » За да демонстрираме някои логически функции, които могат да се използват за решаване на реални проблеми ще разгледаме интелигентна система за подпомагане на зеленчукова борса.
- » Системата трябва да решава задачи като например следните:
 - > Сортиране на зеленчуци по различни признаци, в зависимост от предназначението на произведената продукция.
 - > Прогнозиране търсенето.



Логическо ИЛИ



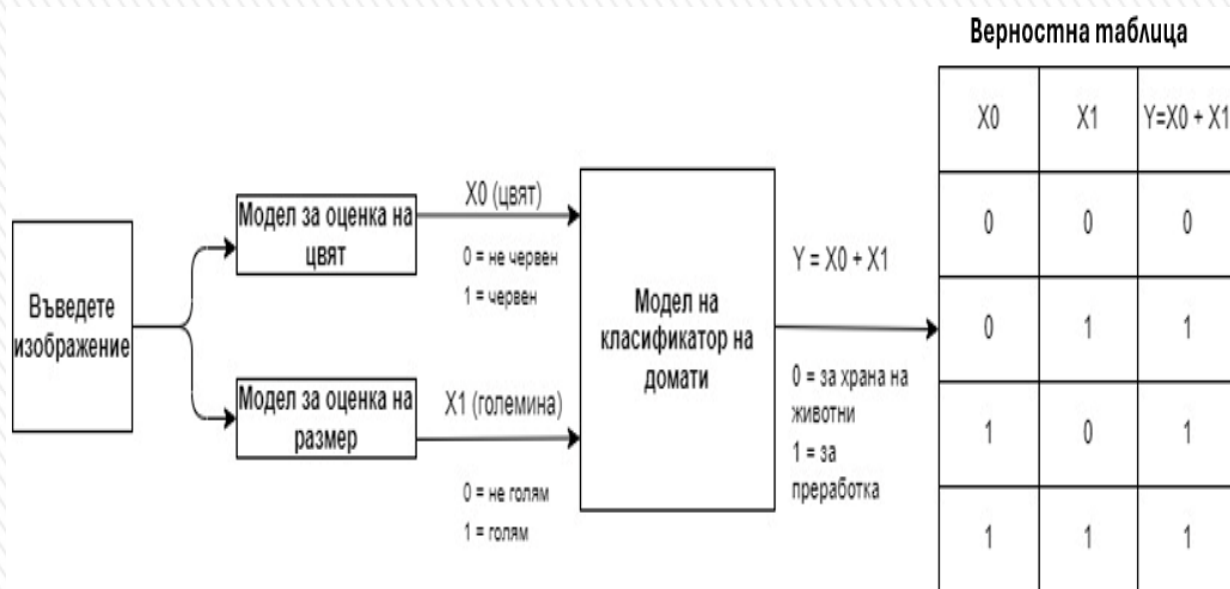
На фигурата е представена схема на решение на задачата за сортиране на домати по цвят и размер, като само зелените и малките ще се използват за храна на животни, а другите отиват за по-нататъшна преработка.

Както се вижда от схемата използват се два оценъчни модела – на цвят и на големина.

Оценъчният модел за цвят връща като резултат двоично решение 0 (не червен) или 1 (червен).



Логическо ИЛИ



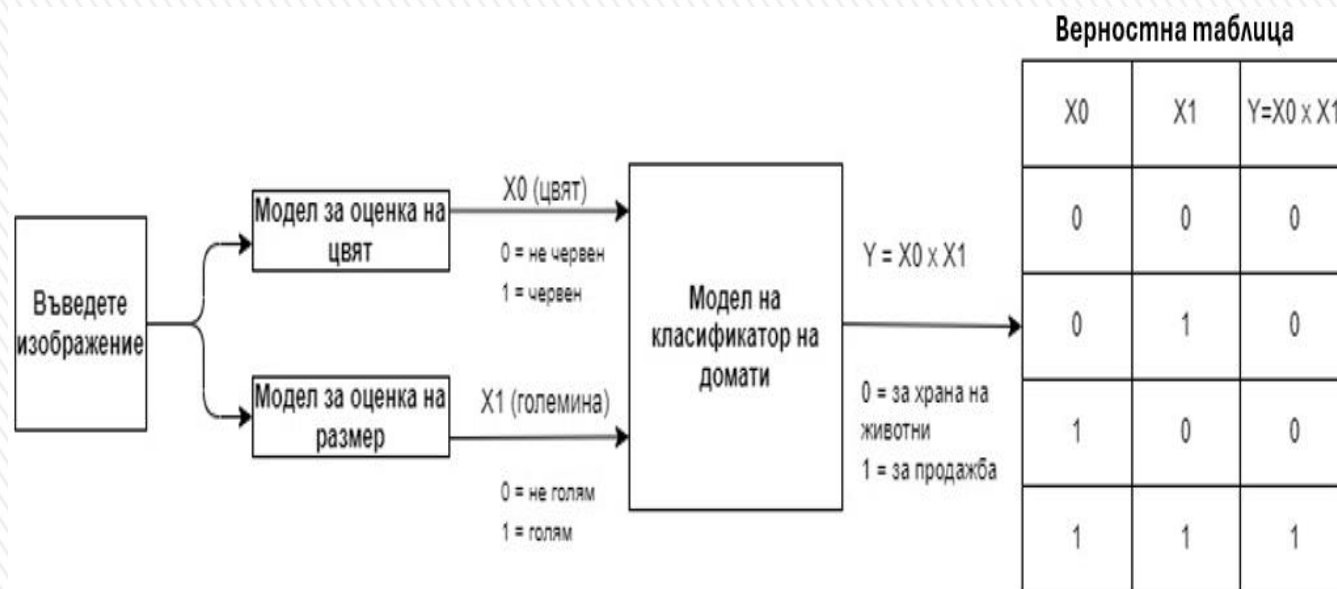
Другият оценител връща съответно двоично решение 0 (не голям) или 1 (голям).

Логическата ИЛИ машина комбинира двоичните решения от двата отделни оценителя.

Може също да се използва логическо ИЛИ с повече от два входа – например при оценката на домати искаме да включим допълнителни характеристики.



Логическо И



В този случай задачата е да се сортират подходящи за продажба домати, т.е. само тези, които са червени и достатъчно големи.

Схемата на решение е дадена на фигурата.

Тук също аналогично на ИЛИ може да се използва логическо И с множество входове.



Логическо НЕ

- » Използването на тази логическа функция се демонстрира със задачата за създаване на програма, която е в състояние да идентифицира зрели домати в дигитални изображения.
- » Да предположим, че имаме на детектор, който е в състояние да разпознава зелени домати и да връща като резултат двоично решение 1 (разпознат зелен домат) или 0 (неразпознат зелен домат).
- » Цялостната система взема изхода от детектора и изпълнява логическа операция НЕ върху него.



Логическо XOR



Верностна таблица

X0	X1	$Y = \neg(X0 \oplus X1)$
0	0	1
0	1	0
1	0	0
1	1	1

Следващата сортировъчна задача, която трябва да решава, е окомплектоване на пакети, съдържащи домати и краставици с подобна големина, т.е. малки краставици с малки домати и големи краставици с големи домати.

Исключващата ИЛИ е подходяща функция за решаване на тази задача.





Благодаря за вниманието!

