

Лекция 12: Невронни мрежи и генетични алгоритми

1 Постановка на задачата

Дадено е множество Ξ от n -мерни вектори \mathbf{X} с компоненти $x_i, i = 1, \dots, n$. Стойностите на компонентите могат да са реални или булеви числа. За всяко \mathbf{X} в Ξ се знае и съответното действие a (**етикет** или **клас** на вектора). Множеството Ξ и етикетите заедно образуват т. нар. **учебно множество**. Задачата е да се намери функция $f(\mathbf{X})$, която да отговаря “приемливо” на елементите на учебното множество, т. е. за колкото може повече вектори в Ξ да поражда етикетите им.

Ако учебното множество е “типично” за предметната област, може да се покаже, че f “вероятно” ще дава “приблизително вярна” стойност и за аргументи извън учебното множество (ще **обобщава**). Точността на функцията може да се оценява с различни методи.

2 Обучение на отделни логически елементи

перцептрон или **адалин** (**адаптивен линеен елемент**): система с единствен прагов логически елемент, определена от **тегла** $\mathbf{W} = (w_1, \dots, w_n)$ и **праг** θ , която отговаря на всеки вход \mathbf{X}

със стойността $f = \begin{cases} 0, & \text{ако } s < \theta \\ 1 & \text{иначе} \end{cases}$ при $s = \mathbf{W} \cdot \mathbf{X} = \sum_{i=1}^n w_i x_i$. Такава система може да се научи да моделира някакво линейно разделение на учебното множество.

(Алтернативно определение чрез **разширени вектори**: векторите са $n + 1$ -мерни, като $w_{n+1} = -\theta$ и $x_{n+1} := 1$; тогава отговорът е $f = \begin{cases} 0, & \text{ако } s < 0 \\ 1 & \text{иначе} \end{cases}$.)

Обучението може да стане, като се минимизира стойността на сумарната **функция на грешката** $\epsilon = \sum_{X_i \in \Xi} (d_i - f_i)^2$, където f_i е полученият отговор на вход X_i , а d_i — очакваният.

Може да се работи и последователно с грешките за отделните входни вектори $\epsilon_i = (d_i - f_i)^2$.

Вместо праговата функция със само две възможни стойности е целесъобразно да се използва сигмоидата $f(s) = \frac{1}{1+e^{-s}}$, $0 < f(s) < 1$. Така между другото може да се оцени размерът на грешката и да се изчисли съответната корекция.

3 Невронни мрежи

Системи от паралелно работещи прагови логически елементи, които могат да моделират нелинейни разделения и/или да изработват повече от две различни стойности. (В крайна сметка представляват математически модел на човешката нервна система, състояща се от много голям брой – от порядъка на 10^{11} – **неврони**, които се възбуждат взаимно, като си предават електрически импулси — алтернатива на последователната класическа компютърна архитектура.) Реализираната функция зависи от броя, теглата и праговете на елементите и от топологията на мрежата (с цикли или без, с един или друг брой слоеве).

Многослойна **нерекурентна** система: елементите в първия слой са **сетивни** (те отговарят за представянето на стимула в мрежата); елементите в слоевете без първия и последния са **скрити**.

За да може да се предположи, че системата отразява някаква закономерност в данните, броят вектори, използвани при обучението, трябва да е по-голям (желателно значително по-голям) от броя на степените свобода (променливи тегла) в мрежата. Това прави задачата нетривиална. (Особено съществено при шумни данни. Иначе системата може да се научи да отразява шума, което би попречило на обобщаването — **пренастройване (overfitting)**.)

Проблеми, свързани с разработването на невронни мрежи:

- много от основните принципи, които управляват обработката на информация в мрежите, са трудни за разбиране;
- сложните взаимодействия между елементите не позволяват прилагането на методи от вида “разделяй и владей”.

Затова възниква въпросът за частично (или пълно) автоматизиране на процеса.

4 Генетично програмиране

Еволюция: създаване на поколения по-добри индивиди посредством

- промяна чрез възпроизводство
- и избирателно оцеляване на част наследниците.

При генетичното програмиране по този начин се “отглеждат” програми.

- Поколение 0 представлява популация от случайни програми, използващи функции, константи и входни данни, които смятаме, че ще трябва.
- Поколение $i + 1$ включва
 - някои от най-добрите представители на поколение i , избрани чрез турнир, върху малка част от които (около 1%) може да се приложи мутация, като случайно избрана част се замени със случайно породен конструкт (по този начин се осигурява възможност да се достигне всяка точка от пространството на търсене и да се излезе от локален минимум),
 - и наследници пак на някои от най-добрите представители на поколение i , получени чрез кръстосване, при което някоя случайно избрана част на бащата замества някоя случайно избрана част на майката.

Методът може да се приложи и върху невронни мрежи, като тяхното обучение се редува с промяна чрез възпроизводство и подбор, а при кръстосването изходните възли на родителите стават скрити възли на наследника.