

Основне академске студије  
Информациони инжењеринг

Основи рачунарске интелигенције

# Перцептрон

(материјали за вежбе)

**1. Увод**

**2. Имплементација перцептрона**

## Перцептрон

основна једнослојна неуронска мрежа

подесна за линеарно сепарабилне класе задатака

ОСНОВНИ ПОДАЦИ

дело Френк Розенблата (енгл. *Frank Rosenblatt*) и сарадника

настанак 1958. године

*Cornell Aeronautical Laboratory*

## Перцептрон

улази

вредности карактеристика, особина

тежина се додељује свакој карактеристици

излаз, активација

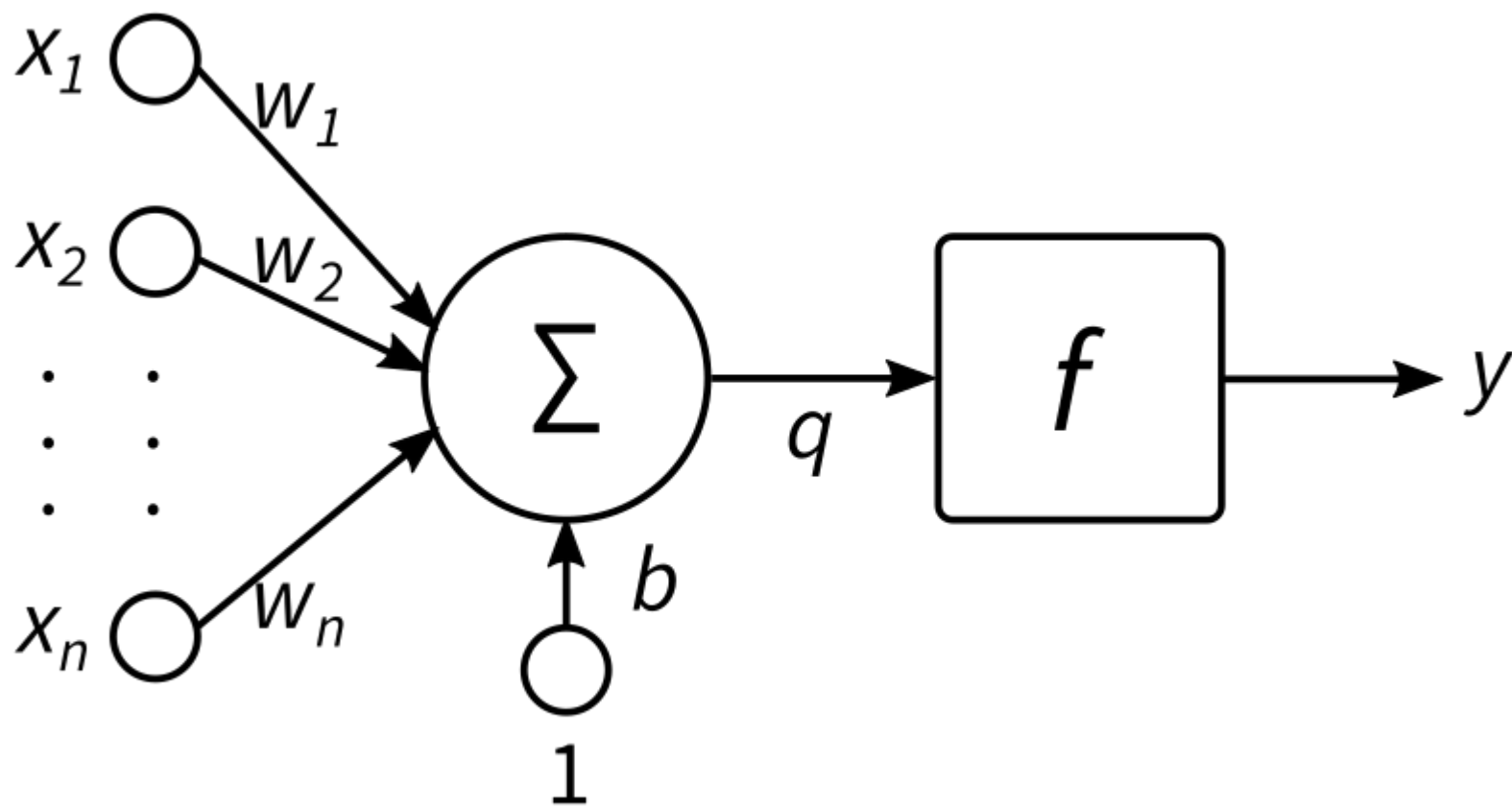
сума тежина и вредности карактеристика

може се додати и праг (енгл. *bias*)

ако је сума већа или једнака 0 излаз је 1, а у супротном излаз је 0

## Перцептрон

модел једног неурона



1. Увод

**2. Имплементација перцептрона**

# Имплементација перцептрона

## Генерисање излазних вредности

пропагација унапред (енгл. *forward propagation*)

$$y = f(Wx + b)$$

функција активације  $f$

1 за аргумент већи од или једнак са 0

0 иначе

## Обучавање перцептрона

подешавање вредности параметара

$$W^{i+1} = W^i + e x^T$$

$$b^{i+1} = b^i + e$$

$i = 0, 1, 2, \dots$

$e$  – грешка

разлика између очекиваног и генерисаног излаза

## Генерисање излазних вредности

помоћу тренутних тежина и прага за сваку вредност из прослеђеног скупа података израчунати вредност суме за добијену вредност суме

ако је вредност већа или једнака 0 функција активације на излаз за сваку вредност из прослеђеног скупа враћа 1

ако је вредност мања од 0 функција активације на излаз за сваку вредност из прослеђеног скупа враћа 0

повратна вредност функције представља скуп 0 и 1  
класификована вредност улаза

## Алгоритам обучавања

измена вредности тежина и прага у више епоха

пример из обучавајућег скупа класификовати помоћу тренутне неуронске мреже

ако је класификација тачна

нема промене матрице тежина и прага

ако је класификација погрешна

мења се матрица тежина

мења се праг

# Имплементација перцептрона

## Модел перцептрона

употребом класа у програмском језику *Python*

класа **Perceptron** представља модел перцептрона

поља класе **Perceptron**

**W** – тежински коефицијенти (тежине)

матрица  $1 \times n$

$[[w_1, w_2, \dots, w_n]]$

иницијализација

на вредност 0

на насумичне вредности

**b** – праг

скалар

иницијализација

на вредност 0

на насумичну вредност

## Модел перцептрона

### методе класе **Perceptron**

**formiranje\_izlaza()** – метода за генерисање излазне вредности над датим улазом

**treniranje()** – метода за обучавање  
подешавања  
подаци  
улазни подаци  
очекивани излазни подаци  
број епоха

**ispitivanje\_performansi()** – метода за проверу исправности класификација

подешавања  
подаци  
улазни подаци  
очекивани излазни подаци

## Подаци

скуп података о резултатима хемијских анализа вина узгајаних у истој регији у Италији добијених од три различите сорте

постоји 178 примерака у скупу и 14 обележја (класа, алкохол, јабучна киселина, пепео, алкалност пепела, магнезијум, укупни феноли, флаваноиди, нефлаваноидни феноли, проантоцијанини, интезитет боје, нијанса, *OD280/OD315* разблажених вина, пролин)

извор скупа података

скуп података *Wine Data Set*

матична Интернет страна

<https://archive.ics.uci.edu/ml/datasets/Wine>

подаци доступни у *CSV* формату

скуп података је креиран 1. 7. 1991. године

за власника скупа података су назначени *Forina, M. et al, PARVUS*

за донатора скупа података је назначен *Stefan Aeberhard stefan@coral.cs.jcu.edu.au*

## Подаци

редови скупа података по класама

1 (59 редова)

2 (71 ред)

3 (48 редова)

посматрање првих 130 редова

подаци раздвојени на класу 1 и класу 2

прва колона је за зависну променљиву

одређивање којој класи вино припада

остале колоне су за независне променљиве

примена код обучавања као и тестирања

## Задатак 1.

Учитати задати скуп података користећи *pandas* библиотеку. Скратити скуп тако да садржи 130 инстанци. Обучити перцептрон над 80% података (обучавајући скуп), а тестирати га над преосталих 20% података (тест скуп). Користити првих седам независних променљивих. Испитати понашање перцептрона!

## Задатак 2.

Урадити задатак 1 али тако да вредности обучавајућег и тест скупа буду нормализоване.

## Задатак 3.

Испробати разна подешавања при обучавању, на пример смањивати број епоха на 200, 120 и 50. Испитати како се мења тачност.

## Литература

Hagan MT, Demuth HB, Beale MH, De Jesús O. Neural Network Design. 2nd Edition. eBook. Internet: <https://hagan.okstate.edu/nnd.html>