Signály a informace

13. cvičení

Principy strojového učení a rozpoznávání

1. **Číselná reprezentace objektů**

- Objekty jsou reprezentovány vektory/maticemi číselnými hodnot - **příznaků**, ty získáme měřením, snímáním signálů, zpracováním signálů, apod.
- Příklady příznakových vektorů/matic: [výška, šířka], [věk, teplota, tlak, tep. frekvence], hodnoty amplitudového spektra ve vektoru délky 128, hodnoty jasu v obrazové matici 32x32, atd.

2. **Učení**

- S využitím **trénovací sady** jsou vytvořeny číselné reprezentace (nebo modely) všech tříd v dané úloze
- Učení bývá většinou **supervizované** („učení s učitelem“), kdy je u každého objektu trénovací sady **známa jeho příslušnost k třídě**
- Opakem je **nesupervizované** („učení bez učitele“), kdy třída objektů **není známa** (specifický typ úloh)

3. **Testování**

- Na objektech **testovací sady** se určí **úspěšnost rozpoznávání**

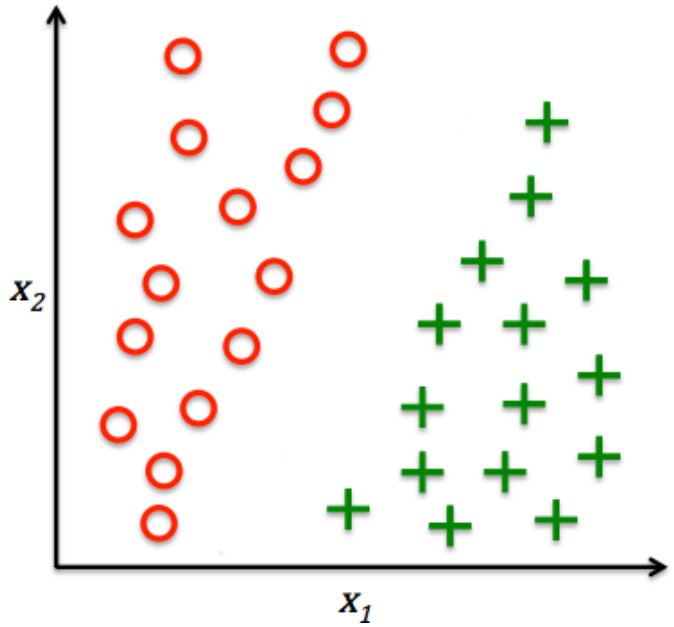
Příznaky a příznakový prostor

Máme-li **N příznaků**, hodnoty příznakového vektoru lze zobrazit v **N-rozměrném příznakovém prostoru**

Objekty každé třídy zabírají vždy určitou část prostoru.

Příklady:

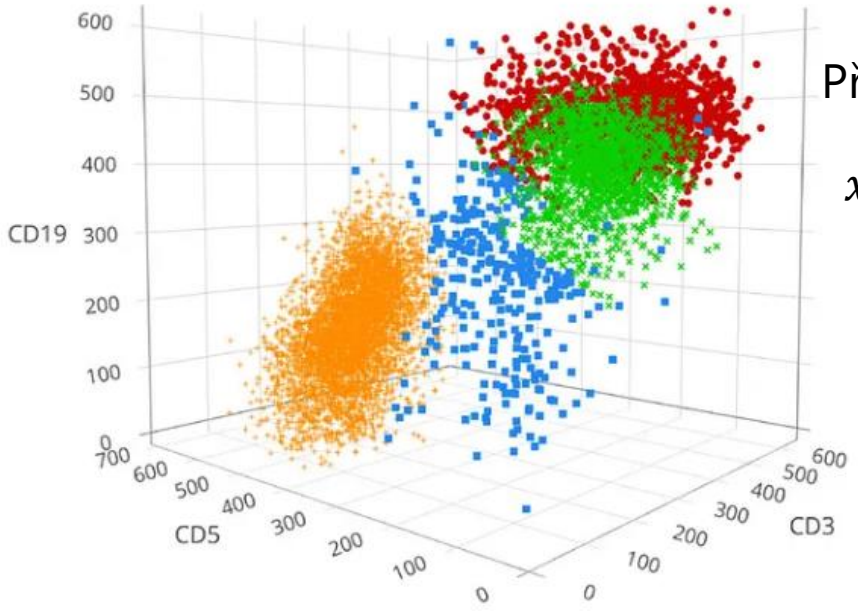
Předměty ze 2 tříd reprezentované 2 příznaky



Příznakový vektor

$x = [x_1, x_2]$

Předměty ze 4 tříd reprezentované 3 příznaky



Příznakový vektor

$x = [x_1, x_2, x_3]$

Parametrické metody rozpoznávání

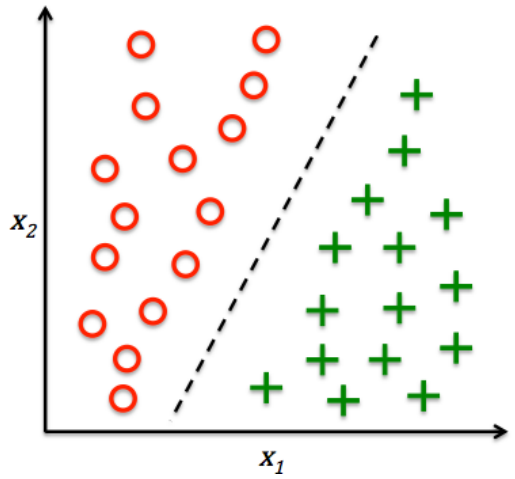
Metoda nejbližšího souseda vs. Parametrické/modelové metody klasifikace

- Metoda **NN vyžaduje určení vzdáleností ke všem objektům** trénovací sady (často výpočetně náročné)

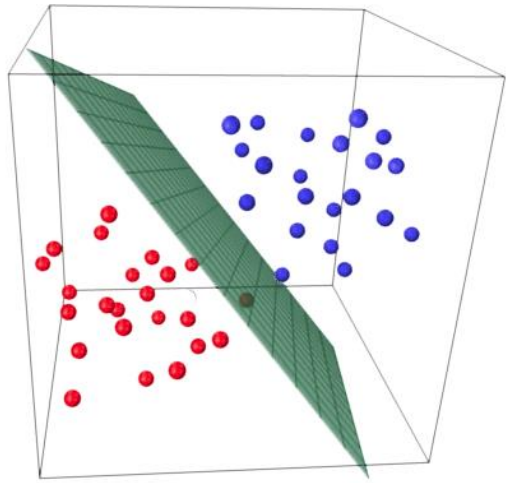
- Efektivnější metody pracují např. **s naučenými modely** (s parametry specifickými pro každou třídu) nebo na principu **rozdělení příznakového prostoru** na podprostory specifické pro každou třídu

Ilustrace:

2D prostor, 2 třídy, rovina rozdělená přímkou



3D prostor, 2 třídy, prostor rozdělený rovinou



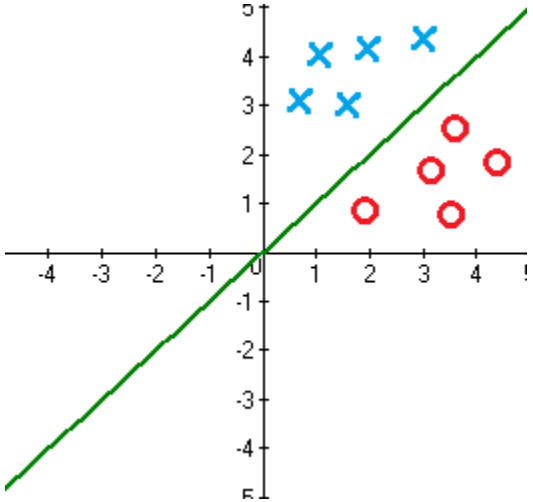
Lineární klasifikace - příklad 2D prostor, 2 třídy

Pokud to rozložení dat v 2D prostoru umožňuje, lze použít **jednoduchý lineární klasifikátor**, který určí **parametry přímky** rozdělující rovinu na 2 **poloroviny specifické** vždy pro jednu třídu

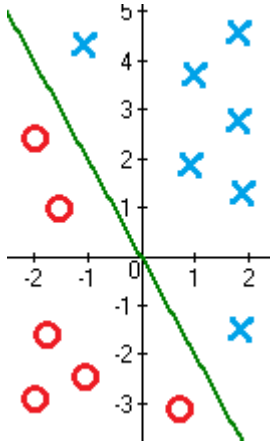
Známe-li rovnici této přímky, můžeme objekty **klasifikovat** na základě toho, zda leží **nad/pod přímkou**.

Příklady dat a rovnic pro rozdělující přímky

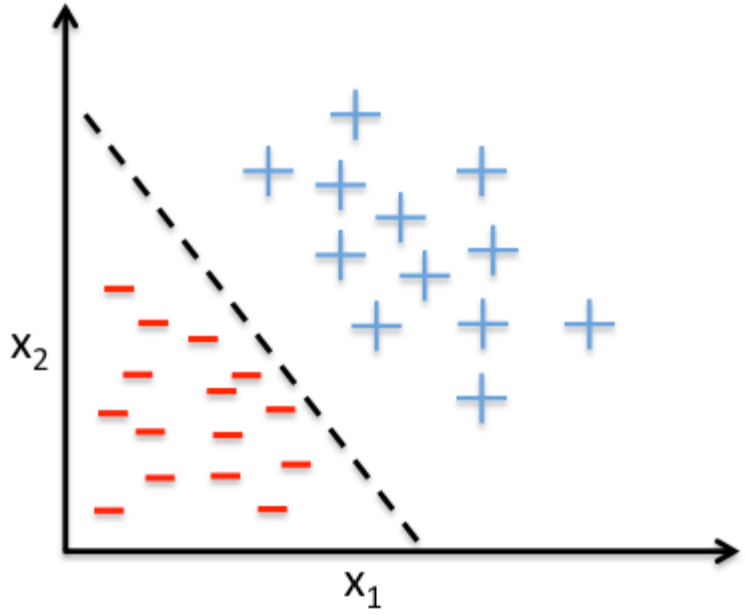
$y = x$



$y = -2x$



$y = -2x + 5$



Lineární klasifikace – rozdělující přímka

Jak funguje lineární klasifikátor pracující s rozdělující přímkou?

Parametrická rovnice přímky: $\mathbf{y = kx + q}$

V prostoru s příznaky označenými x_1 a x_2: $\mathbf{x_2 = kx_1 + q}$

V základním tvaru: $\mathbf{x_2 - kx_1 - q = 0}$

Po zavedení nových proměnných
$w_2 = 1$, $w_1 = -k$ a $w_0 = -q$ $\mathbf{w_2 x_2 + w_1 x_1 + w_0 = 0}$

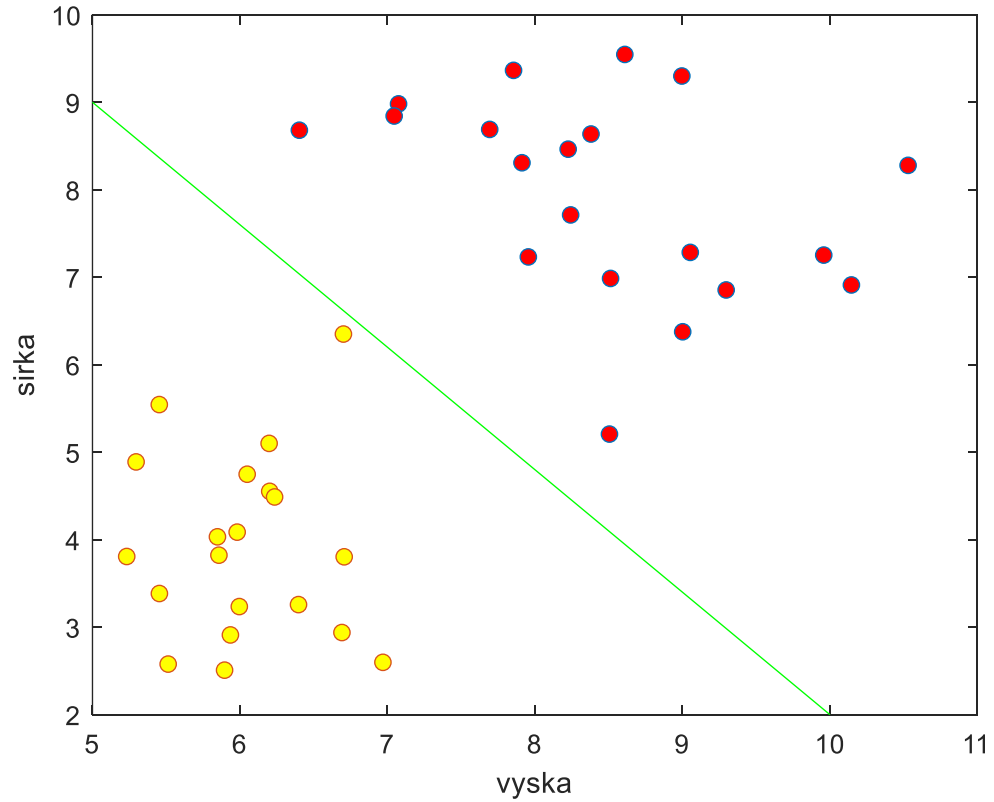
Vektorový zápis vhodný pro implementaci: $\mathbf{x \,.\, w^T = 0}$ kde $w=[w_1\ w_2\ w_0]$ $x=[x_1\ x_2\ 1]$

Známe-li parametry přímky **w**, pak pro objekt s příznaky x_1 a x_2 sestavíme vektor $x=[x_1\ x_2\ 1]$ a platí
$x \,.\, w^T > 0$ objekt je **nad přímkou** a je zařazen do třídy A
$x \,.\, w^T = 0$ objekt je **na přímce** a může být zařazen do jedné ze tříd
$x \,.\, w^T < 0$ objekt je **pod přímkou** a je zařazen do třídy B

Lineární klasifikace – příklad

V úloze se 2 třídami a 2 příznaky máme k dispozici 20 trénovacích vzorků pro každou třídu. Zároveň už známe pozici rozdělující přímky – viz obrázek.

Určete rovnici přímky

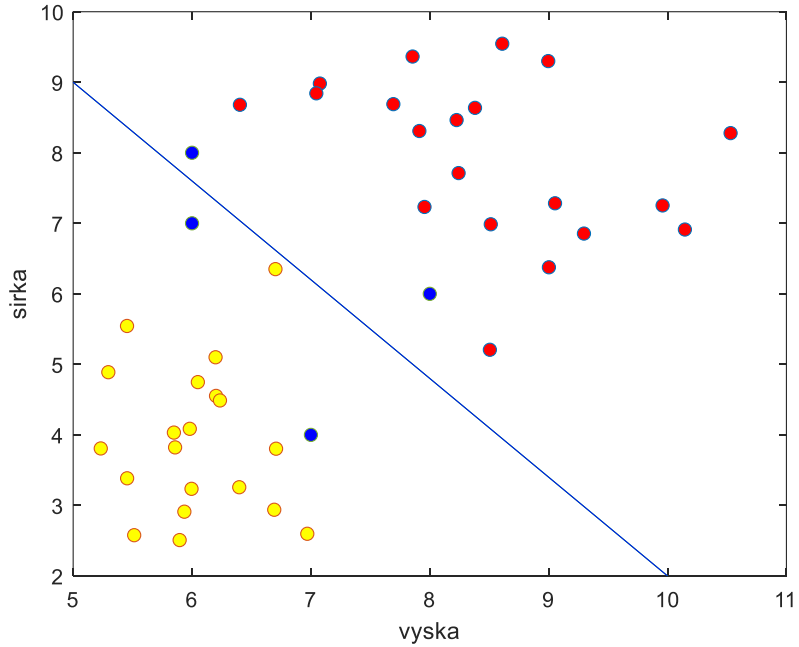


Lineární klasifikace – příklad (pokračování)

V úloze se 2 třídami a 2 příznaky máme k dispozici 20 trénovacích vzorků pro každou třídu. Zároveň už známe pozici rozdělující přímky.

Na základě pozice vůči přímce klasifikujte objekty s příznaky [6, 8] [6, 7] [8, 6] [7, 4].

Řešení:



Perceptron

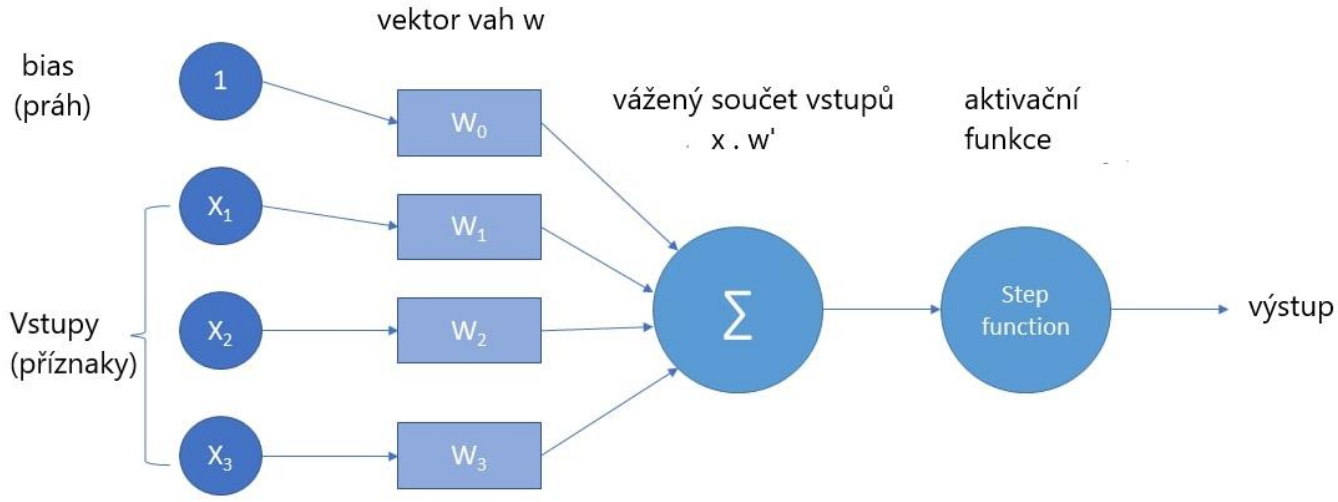
Perceptron byl prvním pokusem o matematickou **simulaci biologického neuronu**, který se rozhoduje na základě průběžného učení na podnětech (datech) - tvůrce Frank Rosenblatt, 1957.
Později se stal základním kamenem většiny **neuronových sítí** (MLP – Multilayer Perceptron, …)

Princip perceptronu: Perceptron přijímá **N vstupních hodnot** (např. příznaků), násobí je **váhovými koeficienty w**, hodnoty **sečte** a po průchodu (většinou nelineární) **aktivační funkcí f** určí **výstupní hodnotu** (např. informaci o zařazení do třídy)

Je to tedy **lineární klasifikátor** pracující na základě rovnic uvedených na předchozích slajdech.

Má **schopnost učit se ze vstupních dat**

Perceptron průběžně (s každým novým vstupním vzorkem) **modifikuje hodnoty váhových koeficientů** tak, aby se dopouštěl **co nejméně chyb při klasifikaci**.



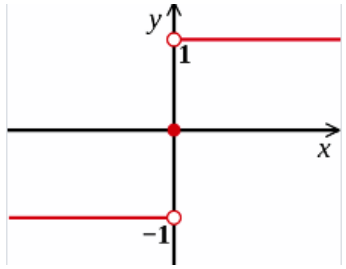
Perceptron

Základní rovnice perceptronu:

$$y = f(x \cdot w^T)$$

Aktiv. funkce **f** bývá nejčastěji funkce **signum**

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{for } x < 0 \\ 0 & \text{for } x = 0 \\ 1 & \text{for } x > 0. \end{cases}$$



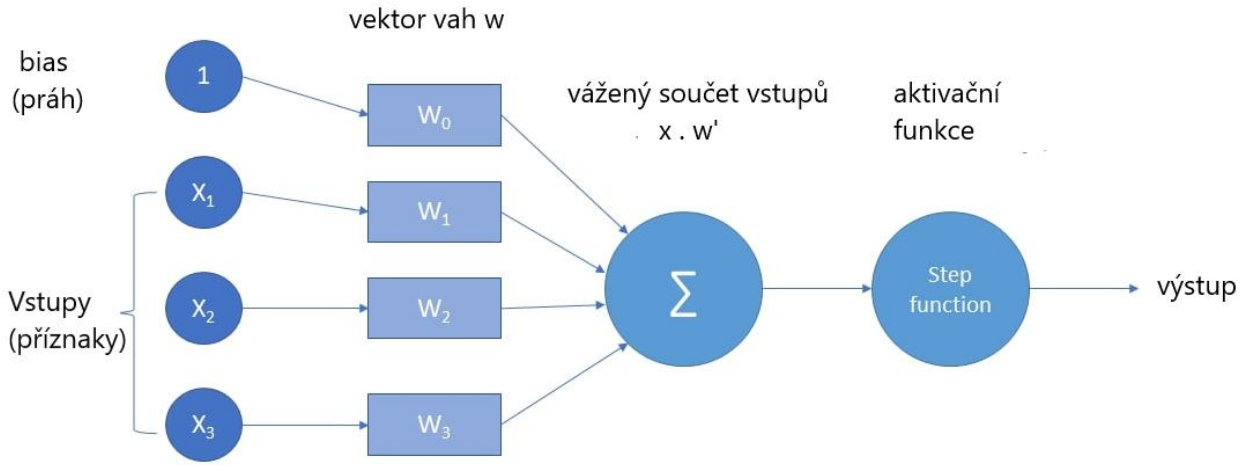
Tedy v Matlabu: **y = sign (x . w^T)**

Interpretace:

perceptron provádí **klasifikaci do dvou tříd**

výstup perceptronu +1 určuje zařazení do třídy A

-1 do třídy B



Perceptron

Princip učení perceptronu:

- Učení spočívá v postupném nastavování takových vah **w**, aby docházelo k minimalizaci klasifikačních chyb.
- Na začátku procesu učení se všechny váhy **w** nastaví na náhodná čísla - nejčastěji v rozmezí (-1, 1)
- Na vstup se postupně přivádějí hodnoty příznakových vektorů **x** reprezentujících objekty ze dvou tříd.
- Pokud je pro daný objekt výstup percetronu **správný** (tj. zařazení do správné třídy), **váhy se nemění**.
- Pokud je pro daný objekt výstup percetronu **nesprávný** (tj. zařazení do nesprávné třídy), **váhy se mění.**

Vztah pro změnu vah: $\mathbf{w} \leftarrow \mathbf{w} + (\mathbf{ys} - \mathbf{y}) \cdot \mathbf{x} \cdot \boldsymbol{\lambda}$

kde **ys** … číslo správné třídy (-1, +1),
y … výstup perceptronu (-1, +1),
x … příznakový vektor daného objektu (který byl nesprávně klasifikován)
λ … faktor učení (číslo většinou menší než 1, které se v průběhu učení snižuje)

Perceptron

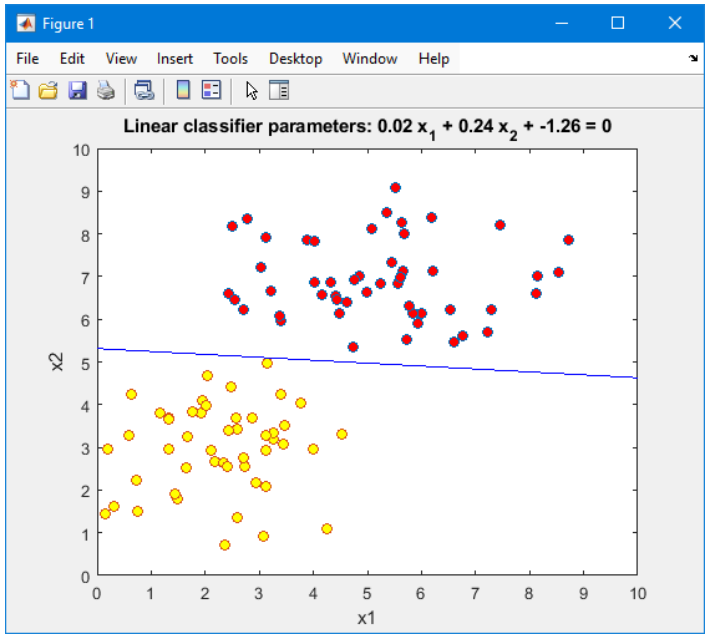
Animace učení perceptronu v Matlabu – program perceptronDemo

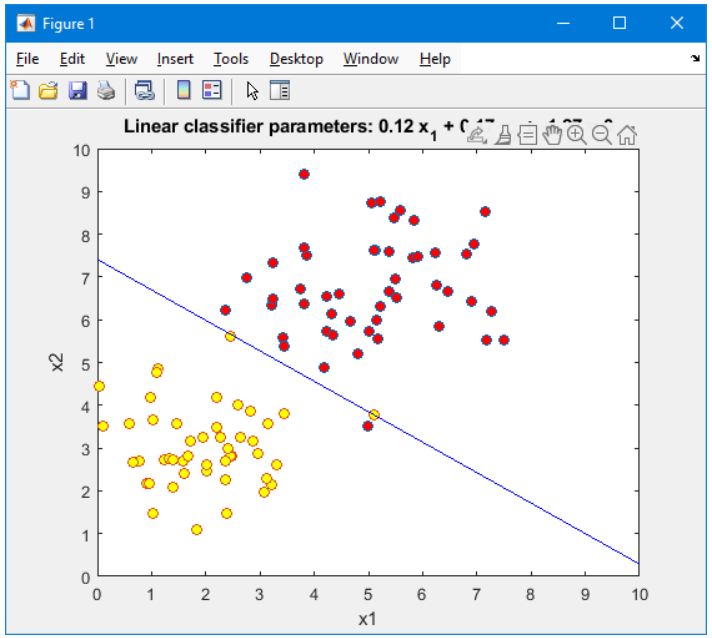
V obrázku jsou zobrazeny **objekty dvou tříd** a neustále **upřesňovaná pozice rozdělující přímky**, jejímiž parametry jsou koeficienty **w** (zobrazené nahoře).

Nová pozice přímky vznikne vždy, když je předložený špatně klasifikovaný.

Ne vždy lze data obou tříd od sebe dokonale oddělit přímkou – v takovém případě jsou tzv. **lineárně neseparabilní** a perceptron se může dopouštět určitého procenta chyb.

Data **lineárně separabilní** vs. **lineárně neseparabilní**





Úloha k odevzdání

Perceptron

Stáhněte si z elearningu program **PerceptronDemo.m.** Několikrát si ho spusťte a sledujte, co a jak dělá.

1. Program provádí následující akce:
   a) do jisté míry náhodně **vygeneruje zadaný počet vzorků třídy A a třídy B**, popsané 2 příznaky x1 a x2,
   b) vytvoří z nich **trénovací sadu** a pomocí ní **natrénuje perceptron** – tj. hledá optimální hodnoty vektoru vah **w**,
   c) v procesu trénování **zobrazuje vzorky** obou tříd a aktuální **pozici rozdělující přímky** danou hodnotami **w**.

2. Proces trénování i vykreslování přímky byl podrobně popsán v dnešní prezentaci a na základě toho byl implementován. Program si **podrobně prostudujte**, abyste mu porozuměli.

3. Vaším úkolem je **doplnit na konec programu** následující (krátké) programové bloky:

   1. Blok nebo funkce s názvem **testPerceptron**, který **použije natrénováné váhy w** a **provede klasifikaci** všech trénovacích vzorků vč. **výpočtu úspěšnosti** (v procentech). Požadovaný blok nebo funkci snadno získáte úpravou funkce **trainPerceptron**.

   2. Blok, který **vygeneruje** a připraví **N (např. 50) testovacích vzorků** podobným způsobem, jako byly vytvořeny trénovací vzorky, tj. s využitím funkce **generateTwoClassData**.

   3. Na tato data aplikujte vámi vytvořený blok/funkci **testPerceptron** a opět **zjistěte úspěšnost** klasifikace.

   4. Výsledky vypište např. takto:

```
disp (['Procento spravne klasifikovanych trenovacich vzorku = ', num2str(trainAccuracy)])
disp (['Procento spravne klasifikovanych testovacich vzorku = ', num2str(testAccuracy)])
```