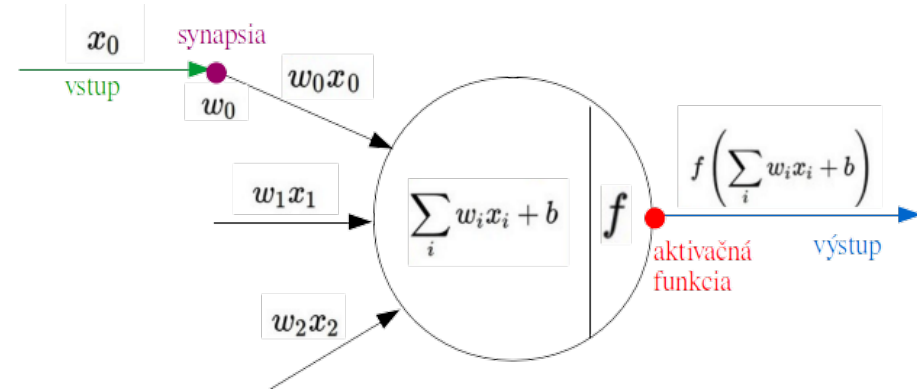
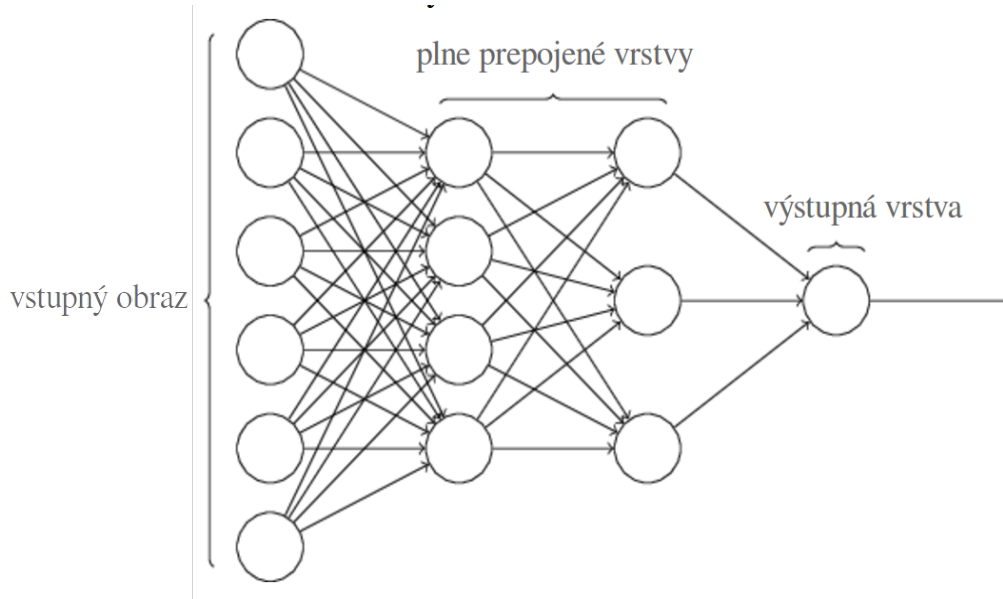


Zadanie 2 – **neurónové siete a regresory**

► Neurónky



Agenda:

1. Príprava množín
2. Architektúra
3. Trénovací proces
4. Vyhodnotenie úspešnosti
5. Sanity checks

►Príprava množín

- Normalizácia
 - Min/max, standardization, L2 normalization
- Príprava/určenie tried
- Rozdelenie na trénovaciu/validačnú/testovaciu
 - Čím viac na trénovanie, tým lepšie
 - Napr. 70/15/15 (80/10/10)
- Vyrovnaný počet trénovacích vzoriek na triedu

► Architektúra

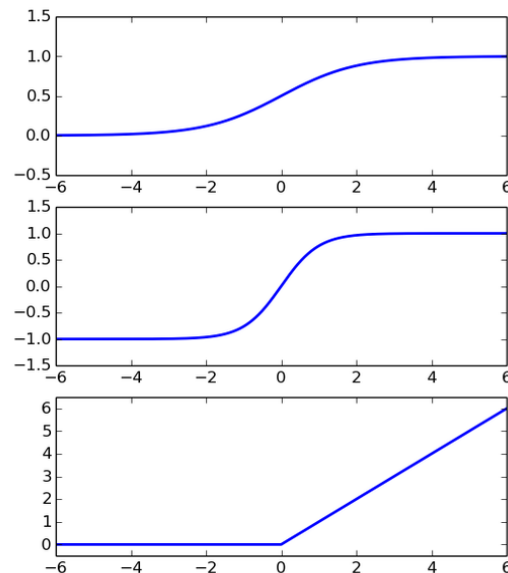
- Počet vrstiev
- Počet neurónov vo vrstvách
- Aktivačná funkcia
- Kritečná funkcia
- Solver

► Architektúra – počet vrstiev/neurónov

- **Vstupná vrstva** - daná veľkosťou jednej vzorky zo vstupu
- Ak nie je lin. separovateľný -> **skryté vrstvy** – nevieme naisto :(- rôzne prístupy:
 1. Čím hlbšie tým lepšie (ale hlboké siete sa ťažko trénujú a sú pomalé)
 2. Počet neurónov medzi veľkosťou vstupu a výstupu
 3. Vzťah medzi počtom vstupných vzoriek a trénovateľnými parametrami -> 10 stupňov voľnosti (ak mám 300 vzoriek môžem trénovať 30 váh)
- **Výstupná vrstva** – jeden neurón (regresia) al. daná počtom kategórii (ak softmax)

► Architektúra - akt./krit. funkcia

- **Aktivačná funkcia** – lineárna/nelineárna
 - Sigmoid, logaritmická -> vanishing gradient, nie centrovaná na nule, pomalá konvergencia
 - Tanh -> vanishing gradient, je centrovaná na nule
 - ReLU -> mŕtve neuróny
- **Kriteriálna funkcia** – podľa typu problému, pre klasif. tried obvykle používame softmax



Sigmoid

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

Hyperbolic Tangent

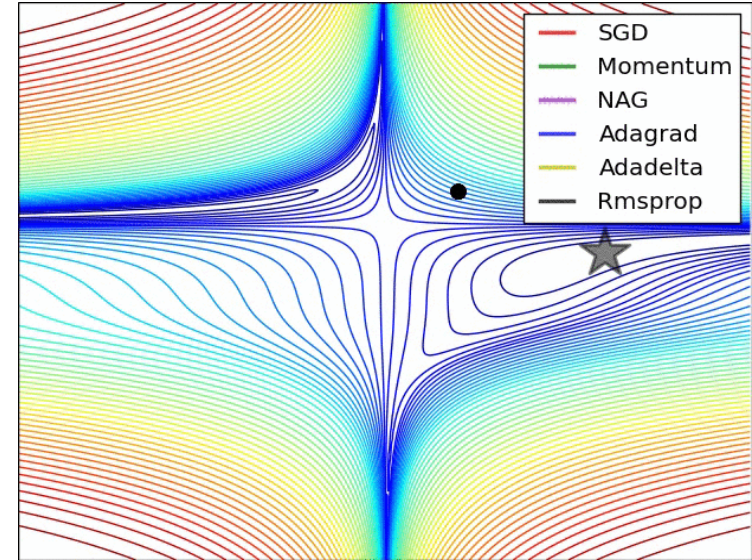
$$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Rectified Linear

$$\phi(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$$

► Architektúra – solver

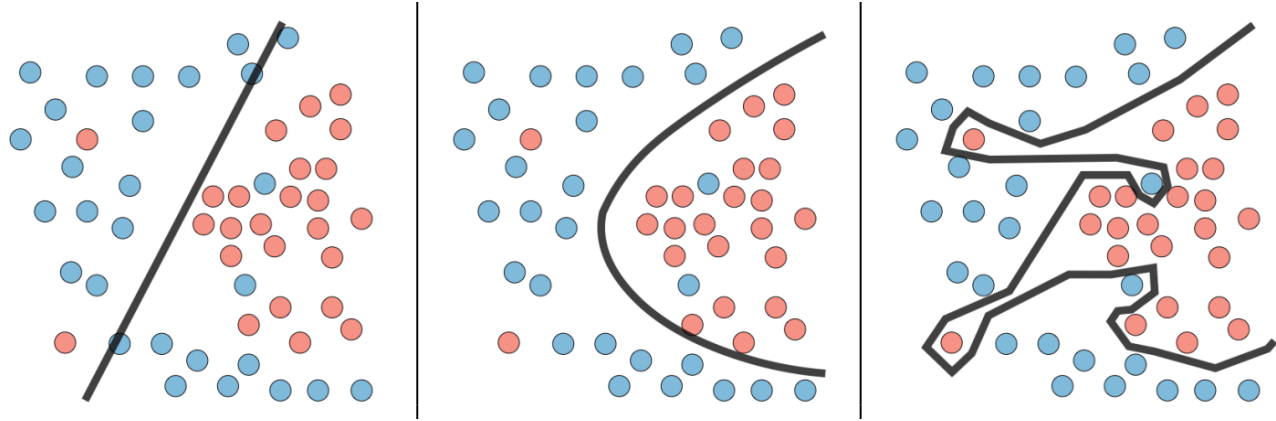
- Vyberáme spôsob úpravy váh a veľkosť parametra rýchlosti učenia
 - Gradientný zostup
 - Stochastický gradientný zostup
 - ADAM
 - Adagrad
- Výber rýchlosti učenia



► Trénovací proces

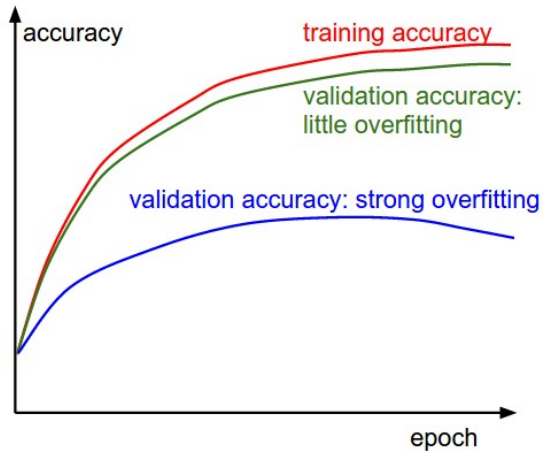
- Prečo má moja sieť zlé výsledky:
 - Zlá architektúra
 - Pretrénovanie
 - Podtrénovanie
 - Zlý parameter rýchlosti učenia
 - ...

► Trénovací proces - pre-/podtrénovanie



! Sledujem kritériálnu funkciu na validačnej množine!

► Trénovací proces - pre-/podtrénovanie



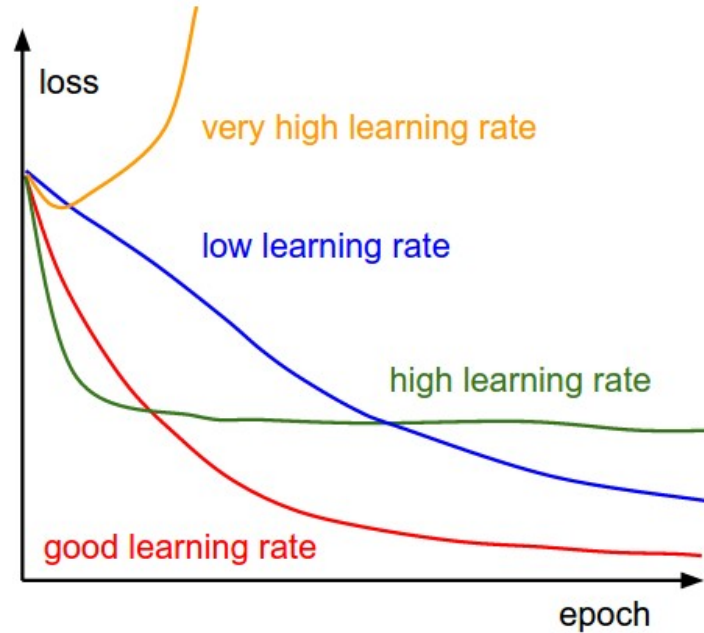
Pretrénovanie:

- Regularizácia
- Viac trénovacích dát
- Menšia sieť
- Early stopping
- Menej príznakov

Podtrénovanie:

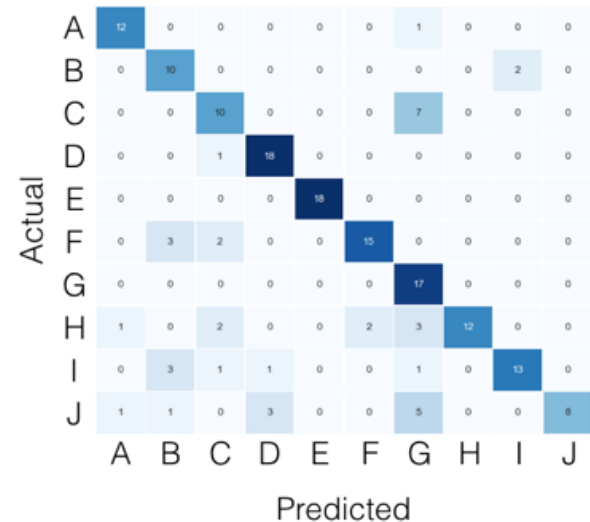
- Zastavovacia podmienka
- Iný model
- Viac príznakov
- Väčšia sieť

► Trénovací proces – parameter rýchlosti učenia



► Techniky vyhodnotenia siete

1. Sledovanie vývoja krit. funkcie
2. Celková úspešnosť
3. Confusion matrix



► Sanity checks

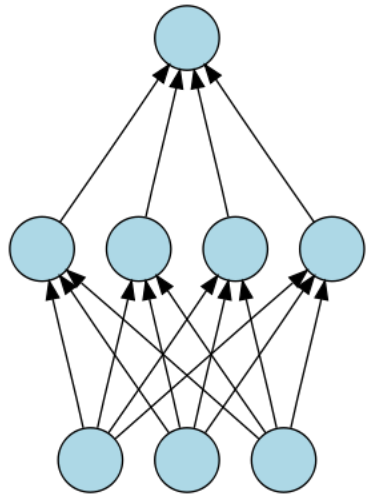
1. V prvom kroku je hodnota kritériálnej funkcie približne rovnaká, ako by mala byť pri náhodnom inicializovaní.
 - softmax $\sim -\ln(1/6) = 1.79$
2. Zvyšovanie regularizácie by malo zvýšiť hodnotu kritériálnej funkcie.
 - ... regularizácia pri ďalšom zadaní
3. Na maličkovej podmnožine trénovacej množiny viete sieť pretrénovať.
 - viete dosiahnuť hodnotu krit. f. rovnú (blízku) nule.

Regresory

► Regresory

- Neurónová sieť
- RBF sieť
- Podporné vektory

► RBF siet'



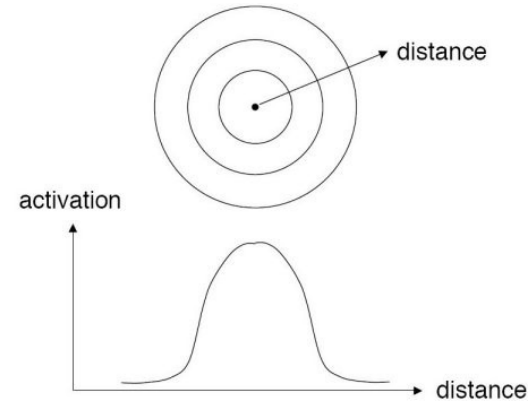
Output y

Linear weights

Radial basis functions

Weights

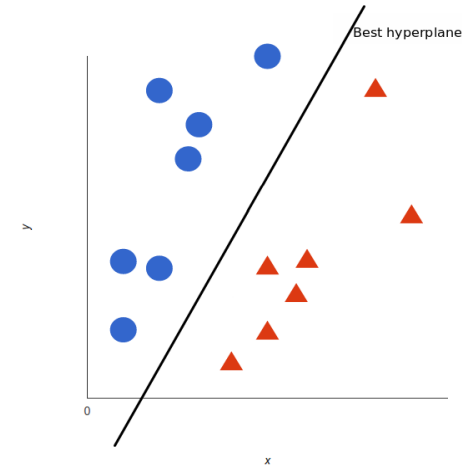
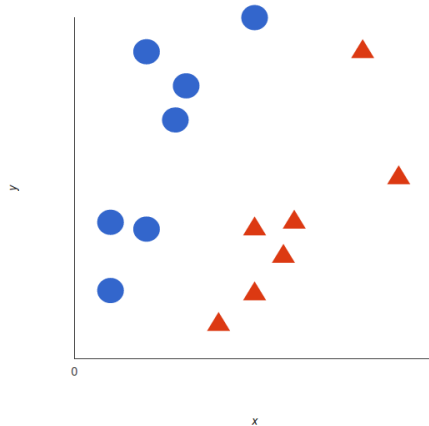
Input x



$$\phi(r) = \exp(-r^2/2\sigma^2)$$

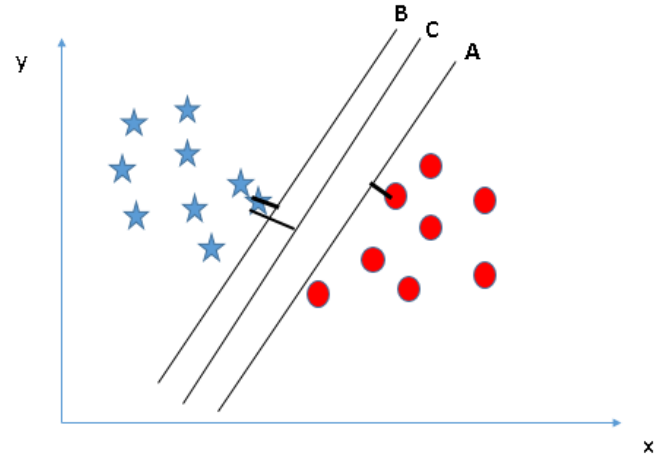
► Podporné vektory

- Vstup: označené body v priestore
- Výstup: hyperroviny oddelujúce triedy (klasifikácia) / približnú funkciu (regresia)



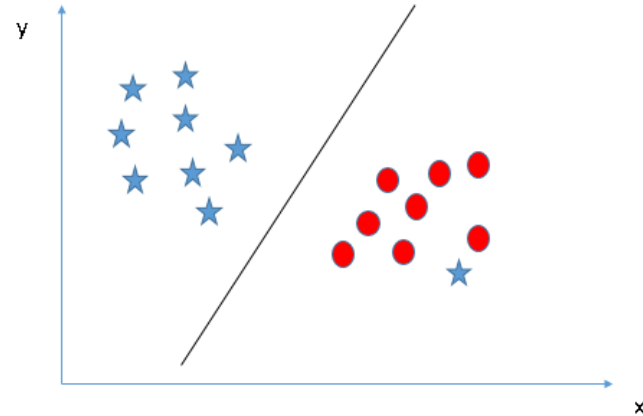
► Podporné vektory - základné koncepty

- Margin
- Outliers
- Nelineárne hyperroviny



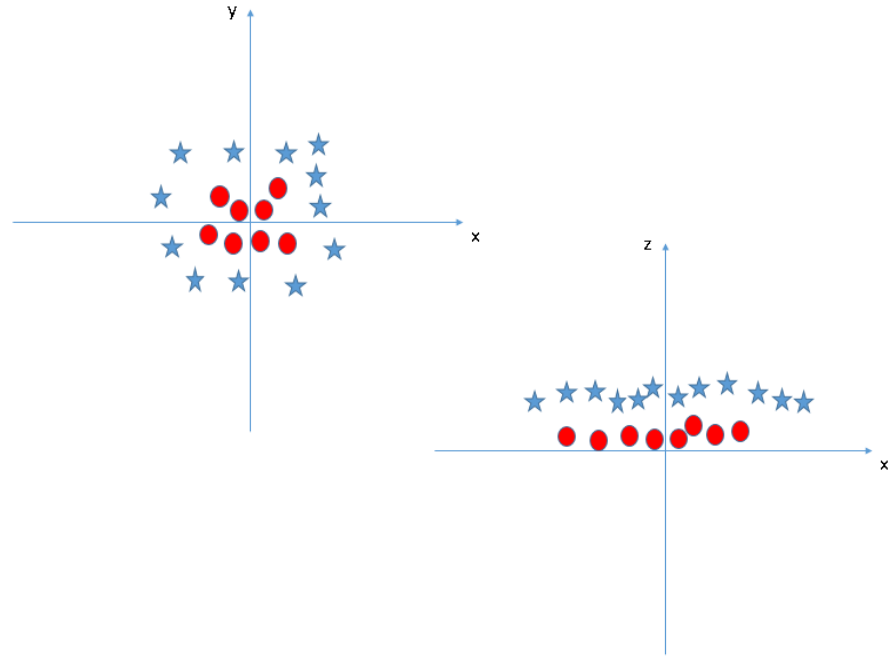
► Podporné vektory - základné koncepty

- Margin
- Outliers
- Nelineárne hyperroviny



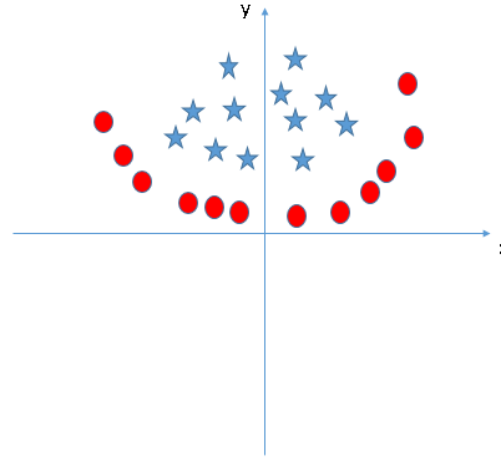
► Podporné vektory - základné koncepty

- Margin
- Outliers
- Nelineárne hyperroviny



► Podporné vektory - základné koncepty

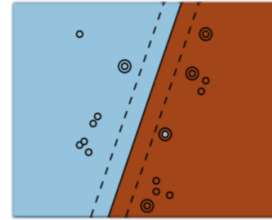
- Margin
- Outliers
- Nelineárne hyperroviny



► Podporné vektory - hyperparametre

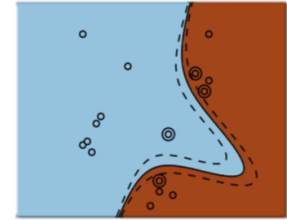
- Druh kernelu
- Hyperparameter C
- Hyperparameter γ
- Hyperparameter stupeň

Linear Kernel



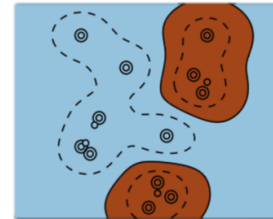
C hyperparameter

Polynomial Kernel



C plus gamma, degree and coefficient hyperparameters

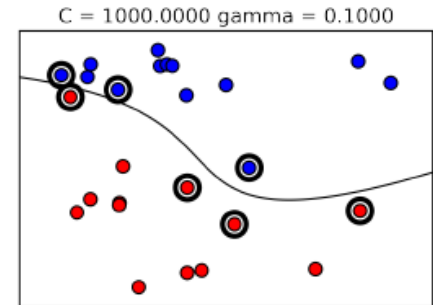
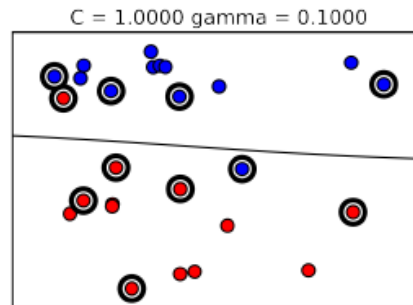
RBF Kernel



C plus gamma hyperparameter

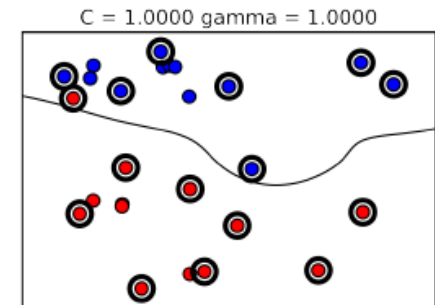
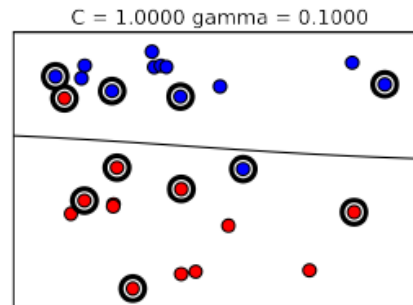
► Podporné vektory - hyperparametre

- Druh kernelu
- Hyperparameter C
- Hyperparameter gamma
- Hyperparameter stupeň

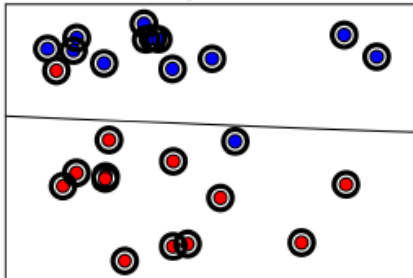


► Podporné vektory - hyperparametre

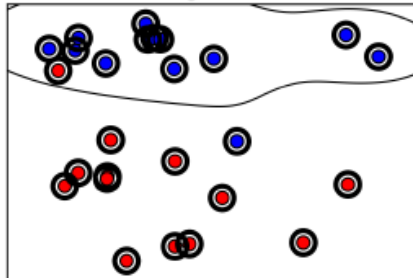
- Druh kernelu
- Hyperparameter C
- Hyperparameter γ
- Hyperparameter stupeň



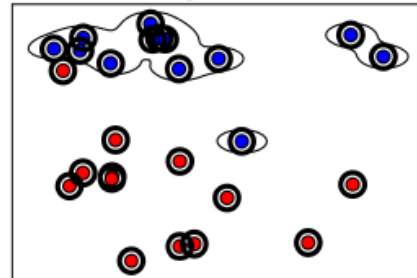
$C = 0.1000$ $\gamma = 0.1000$



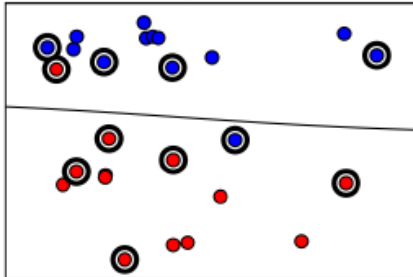
$C = 0.1000$ $\gamma = 1.0000$



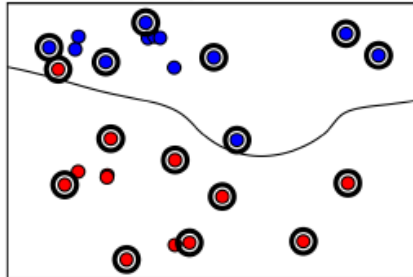
$C = 0.1000$ $\gamma = 10.0000$



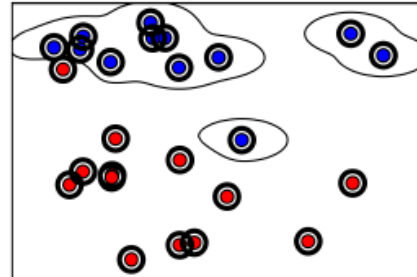
$C = 1.0000$ $\gamma = 0.1000$



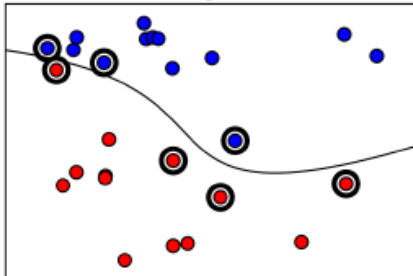
$C = 1.0000$ $\gamma = 1.0000$



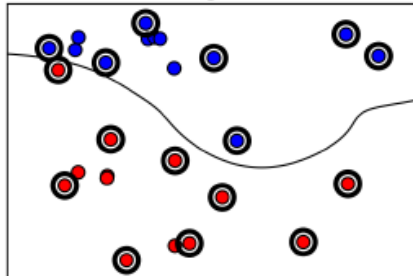
$C = 1.0000$ $\gamma = 10.0000$



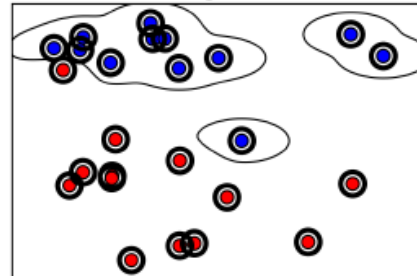
$C = 1000.0000$ $\gamma = 0.1000$



$C = 1000.0000$ $\gamma = 1.0000$

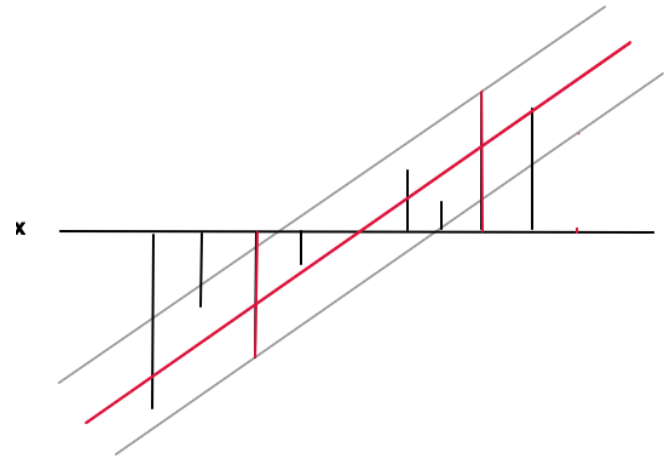


$C = 1000.0000$ $\gamma = 10.0000$



► Podporné vektory + regresia

- Regresia – zmena iba v kriteriálnej funkcii -> teraz chcem mať body čo najbližšie ku nájdeným vektorom



► Vyhodnocovanie regresie

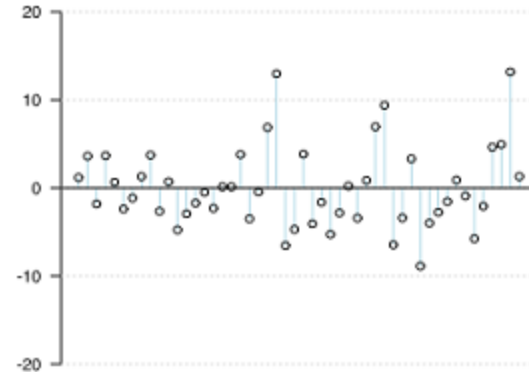
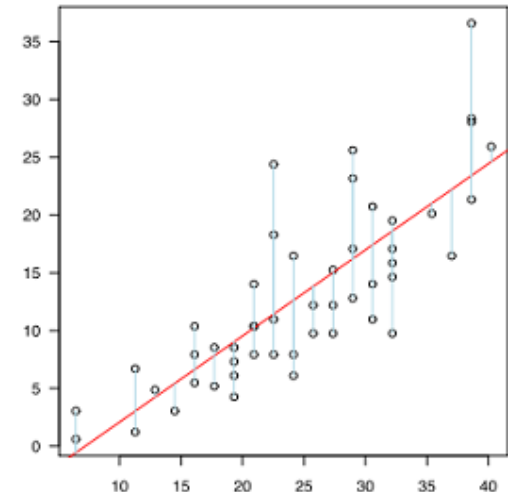
- Štatisticky

- Mean squared error

- R^2
$$\hat{R}^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}$$

- Grafom

- Residual plot





Priestor na otázky