

Supervised learning „učenie s učiteľom“ 3. časť

Jozef Jakubík

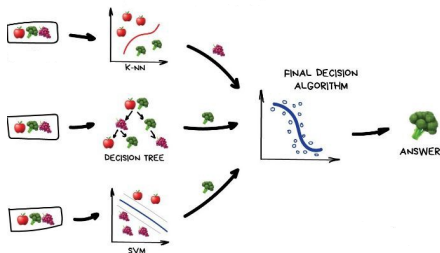
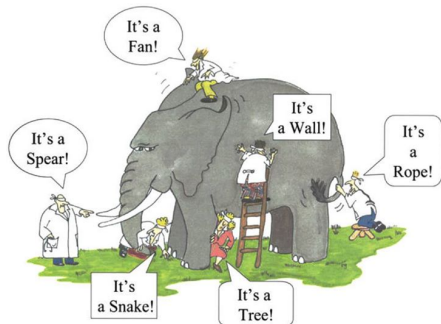
18. novembra 2021

Seminár UM

Ensemble learning

Ensemble learning

Rozhodnutie klasifikácie nie je výsledkom jedného „základného“ modelu (base model), ale kombinácia výsledkov viacerých základných modelov rovnakého typu (trénovaných na rôznych dátach) alebo rôznych typov základných modelov.

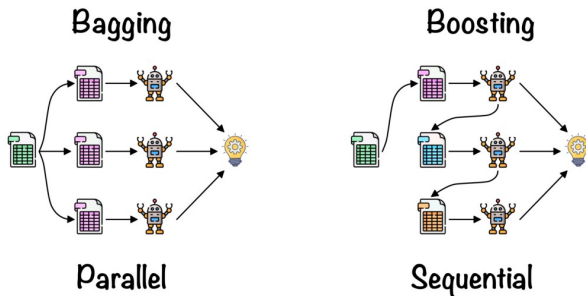


<https://www.patheos.com/blogs/driventoabstraction/2018/07/blind-men-elephant-folklore-knowledge/>

<https://towardsdatascience.com/ensemble-learning-from-scratch-20672123e6ca>

Ensemble learning - delenie

- Bagging - „paralelné“ - trénovacie vzorky pre základné modely sa vyberajú náhodne rovnomerne (každý element má rovnakú pravdepodobnosť výberu) zo všetkých dát.
- Boosting - „sériové“ - trénovacia vzorka sa vyberá na základe výsledkov predchádzajúceho klasifikátora (zle zaklasifikované dáta majú väčšiu šancu dostať sa do trénovacej vzorky).



Ensemble learning - funguje to?

Ensemble learning vychádza z myšlienky bootstrapu (opakované odhadovanie hľadaného parametra na základe náhodne (s opakovaním) vybraných simulovaných výberov) a v praxi majú ním dosiahnuté výsledky lepšie vlastnosti.

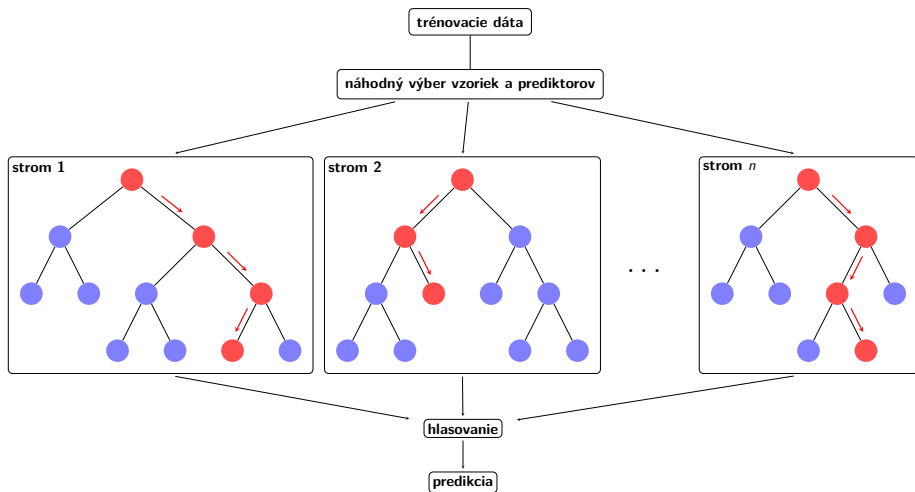
Our results demonstrate that a Bagging ensemble nearly always outperforms a single classifier. Our results also show that a Boosting ensemble can greatly outperform both Bagging and a single classifier. However, for some data sets Boosting may show zero gain or even a decrease in performance from a single classifier.

[Opitz and Maclin, 1999]

[Opitz and Maclin, 1999] Opitz, D. and Maclin, R. (1999). [Popular ensemble methods: An empirical study](#). Journal of artificial intelligence research 11, 169–198

Bagging - Bootstrap Aggregating

Random Forest:



Random Forest - MATLAB

RF = TreeBagger(K,X,Y)

K - počet stromov v lese

X - $n \times p$ matica dát

Y - $n \times 1$ vektor, zaradenie n pozorovaní do tried

● voliteľné parametre

- Method - 'classification' alebo 'regression'. Regression vyžaduje numerické hodnoty v Y
- NumPredictorsToSample - počet náhodne vybraných prediktorov pre každý strom
- InBagFraction - podiel náhodne (s opakovaním) vybraných tréningových vzoriek pre každý strom
- je možné použiť aj parametre týkajúce sa klasifikačných stromov z funkcie *fitctree*

RF - natrénovaný model

Váhy pri náhodnom výbere trénovacích dát pre nasledujúci základný model sú upravené podľa výsledkov aktuálneho základného modelu.

Pravdepodobnosť pre zle zaklasifikované dáta sa zvýši a pravdepodobnosť pre správne zaklasifikované dáta sa zníži.

- AdaBoost (Adaptive Boosting)
- Gradient Boosting
- XGBoost (Extreme Gradient Boosting) - implementácia pre C++, Python, R, Java, Scala, Julia na <https://xgboost.ai/>.

Ada = fitcensemble(X,Y)

X - $n \times p$ matica dát

Y - $n \times 1$ vektor, zaradenie n pozorovaní do tried

- voliteľné parametre

- Method - 'AdaBoostM1' pre binárnu klasifikáciu alebo 'AdaBoostM2' pre viac klás.
- Learners - 'discriminant', 'knn', 'tree'
- LearnRate - (0,1]

Ada - natrénovaný model

- **výhody:**

- v praxi lepšie výsledky ako základné modely
- veľmi užitočné pri vysokom počte dát s mnohými premennými

- **nevýhody:**

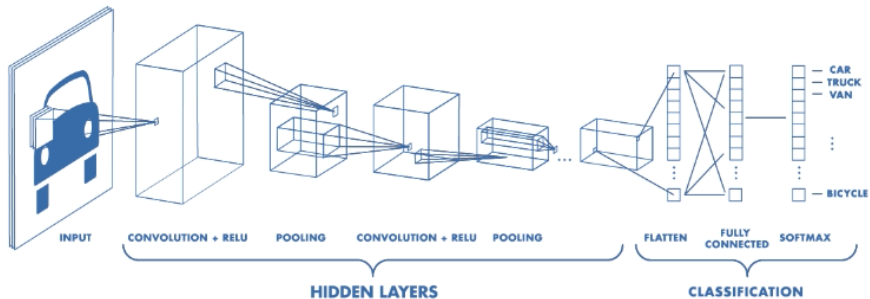
- náročnejšia interpretácia

Neurónové siete

- nelineárna regresia
- klasifikácia
- spracovanie obrazu
- spracovanie jazyka
- reinforcement learning

Spracovanie obrazu - Convolutional Neural Network (CNN)

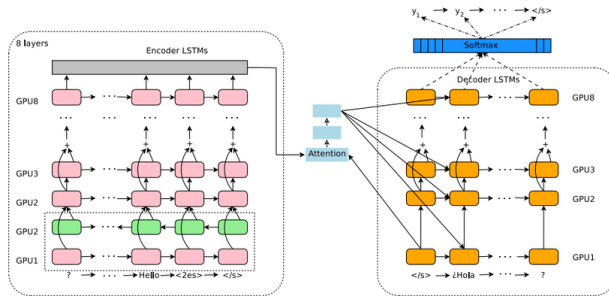
- využívane v prípade viacrozmerých závislostí v dátach (2d - obrázky)



<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Spracovanie jazyka - Recurrent neural network (RNN), Long short-term memory (LSTM), Attention

- využívané v prípade dát, ktoré nemajú pevne danú dĺžku (text - rôzna počet slov vo vete)



Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K. et al. (2016). [Google's neural machine translation system: Bridging the gap between human and machine translation](https://arxiv.org/abs/1609.08144). arXiv preprint arXiv:1609.08144

AlphaGo 4 - 1 Lee Sedol (2016) - umelá inteligencia poráža 18-násobného majstra sveta v Go.

AlphaStar (2019) - umelá inteligencia úspešne súperí s najlepšými hráčmi Starcraftu II.

`https:`

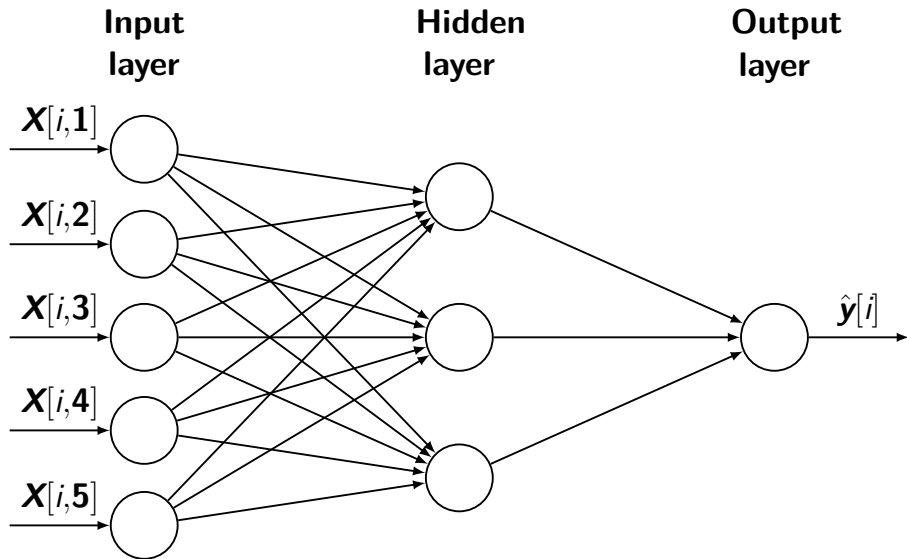
`//deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii`

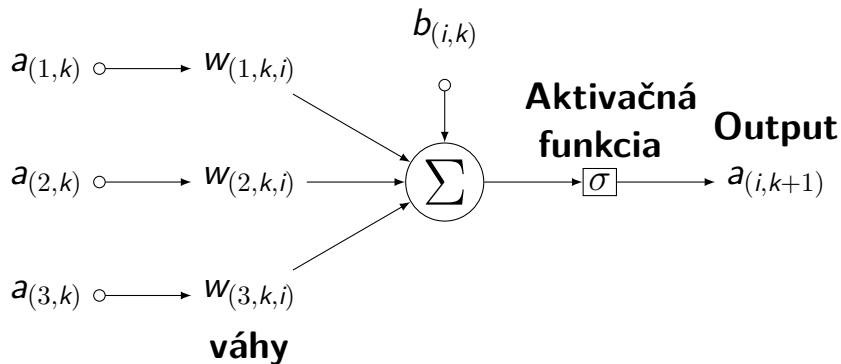
X - $n \times p$ matica dát

Y - $n \times 1$ vektor - závislá premenná (regresia) alebo zaradenie n pozorovaní do tried (klasifikácia)

Problém sa nelíši v štruktúre neurónovej siete, len v účelov funkcii.

Neurónová sieť





Neurón

$$a_{(i,k+1)} = \sigma \left(b_{(i,k)} + \sum_{j=1}^{n_k} w_{(j,k,i)} a_{(j,k)} \right)$$

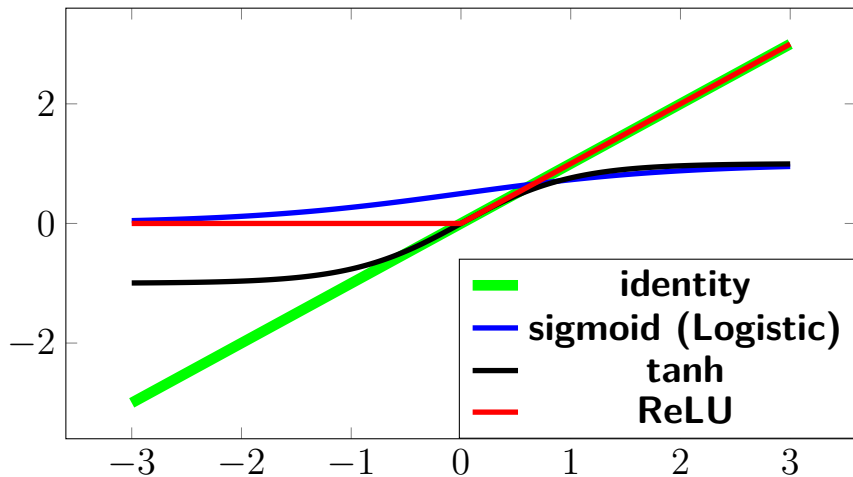
Vrstva

$$\mathbf{a}_{k+1} = \sigma (\mathbf{b}_k + \mathbf{W}_k^T \mathbf{a}_k)$$

Sieť

$$\mathbf{a}_{K+1} (= \hat{\mathbf{y}}[i]) = \sigma (\dots \sigma (\mathbf{b}_2 + \mathbf{W}_2^T \sigma (\mathbf{b}_1 + \mathbf{W}_1^T \mathbf{a}_1 (= \mathbf{X}[i, :]))))$$

Aktivačná funkcia - σ



Účelová funkcia - regresia - Mean Squared Error (MSE):

$$C = \frac{1}{n} \|\mathbf{y} - f(\mathbf{X})\|_2^2$$

Účelová funkcia - klasifikácia - Cross-Entropy:

$$K = - \sum_{c=1}^M y_{o,c} \log(p_{o,c})$$

M - počet tried

y - binárny indikátor (0 or 1) či pozorovanie o patrí do triedy c

p - predikovaná pravdepodobnosť, že pozorovanie o patrí do triedy c

Gradientný postup minimalizácie

$$\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, \dots] \ \& \ \mathbf{b} = [\mathbf{b}_1, \mathbf{b}_2, \dots]$$

$$[\mathbf{W}, \mathbf{b}]^{k+1} = [\mathbf{W}, \mathbf{b}]^k - \gamma \nabla C([\mathbf{W}, \mathbf{b}]^k)$$

Backpropagation [**Rumelhart et al., 1986**]

Stochastic gradient descent - SGD [**Kiefer and Wolfowitz, 1952**]

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). [Learning representations by back-propagating errors](#). nature 323, 533

[Kiefer and Wolfowitz, 1952] Kiefer, J. and Wolfowitz, J. (1952). [Stochastic estimation of the maximum of a regression function](#). The Annals of Mathematical Statistics 1, 462–466

`https://playground.tensorflow.org/`

NN = fitrnet/fitcnet(X,Y)

X - $n \times p$ matica dát

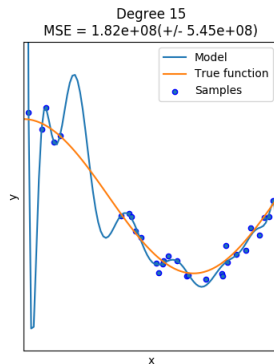
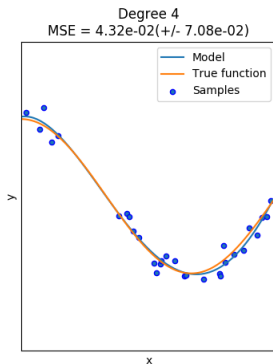
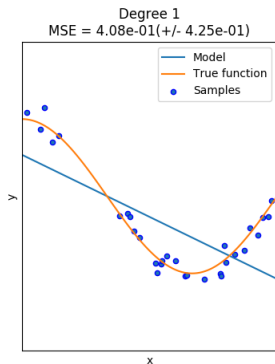
Y - $n \times 1$ vektor - závislá premenná (regresia) alebo zaradenie n pozorovaní do tried (klasifikácia)

- voliteľné parametre

- LayerSizes - počet neurónov na vrstvách (napr.: [100 25 10])
- Activations - 'relu' (default), 'tanh', 'sigmoid'

NN - natrénovaný model

Neurónové siete - počet parametrov



train, test a validation set

train set - množina dát, ktorá sa použije pri tréňovaní modelu

validation set - množina dát na základe ktorej ladíme hyperparametre (napr.: počet skrytých vrstiev neurónovej siete)

test set - množina dát na ktorej meriame presnosť plne špecifikovaného modelu

- **výhody:**

- široké uplatnenie

- **nevýhody:**

- náročnejšia interpretácia
- nájdenie „správneho“ modelu môže byť komplikované

Otázky ?