University of Ljubljana
Faculty of Mathematics and Physics

12. naloga
Strojno učenje

Ma-Fi praktikum 2021/2022

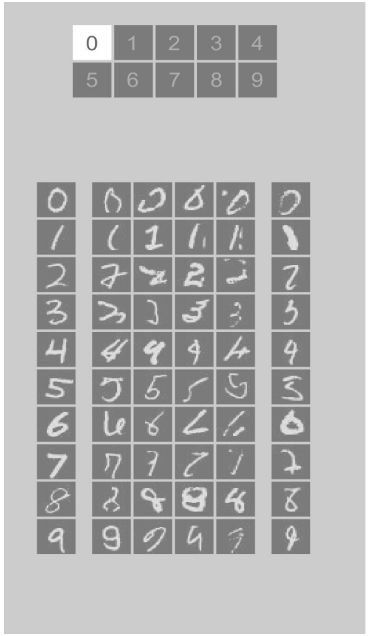
Razvoj strojnega učenja

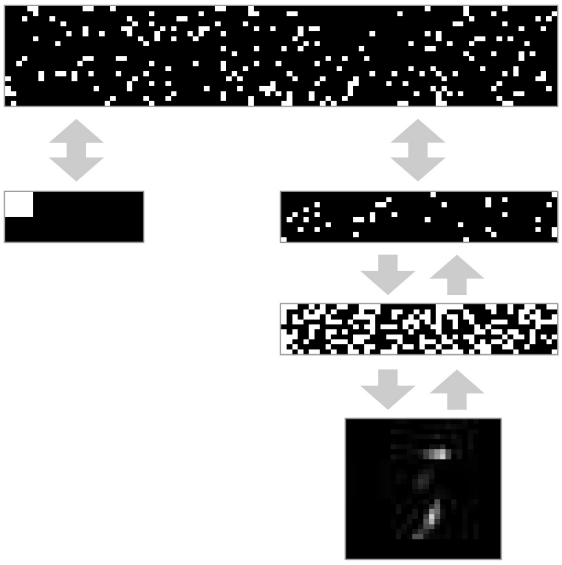
University *of Ljubljana*
Faculty *of Mathematics and Physics*

↳ Sama (matematična) logika je preprosta, izziv je bil (kot na primer v teoriji kaosa) možnost izračuna posameznih scenarijev. **Razvoj računalnikov je torej tu glavni dejavnik!**

↳ Večina teorije ze pred 1990, vendar takrat računalniki še preveč šibki za zahtevne procedure strojnega učenja (recimo milijoni podatkov ....).

↳ Leta 2006 Geoffrey Hinton *et al.* objavijo članek, kako lahko naučimo nevronsko mrežo prepoznavanja pisanih črk s fantastično natančnostjo (>98%). To metodo so poimenovali "Deep Learning."





A fast learning algorithm for deep belief nets *

Geoffrey E. Hinton and **Simon Osindero**
Department of Computer Science University of Toronto
10 Kings College Road
Toronto, Canada M5S 3G4
{hinton, osindero}@cs.toronto.edu

Yee-Whye Teh
Department of Computer Science
National University of Singapore
3 Science Drive 3, Singapore, 117543
tehyw@comp.nus.edu.sg

AlexNet in GPU - DNN

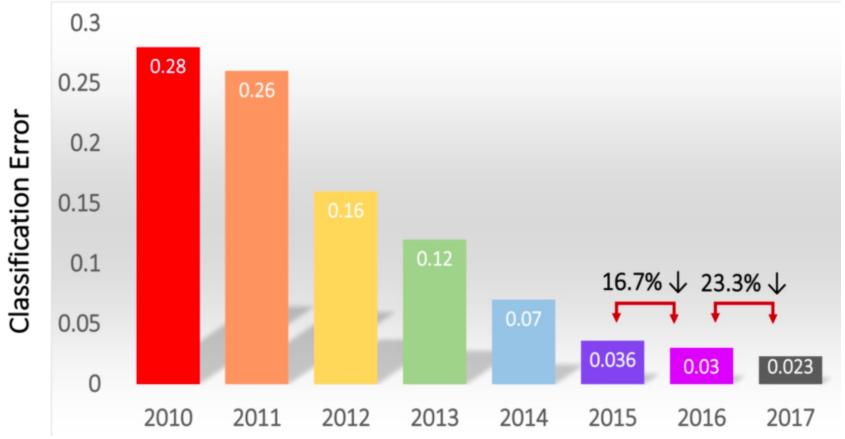
University of Ljubljana
Faculty of Mathematics and Physics

↳ Preboj v uporabi GPU v praksi: AlexNet (tekmovanje Image Net leta 2012):

↳ Resna implementacija Deep Neural Networks (DNN).

The handwriting recognition network discussed above had two layers, 25 neurons, and almost 12,000 parameters. AlexNet was vastly larger and more complex: eight trainable layers, 650,000 neurons, and 60 million parameters.

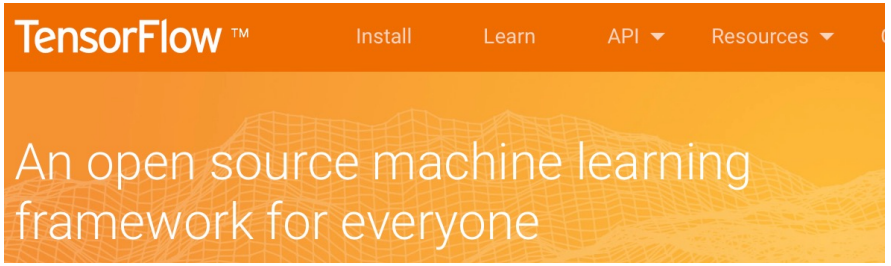
Training a network of that size required massive computing power, and AlexNet was designed to take advantage of the massive parallel computing power provided by modern GPUs. The researchers figured out how to divide the work of training their network across two GPUs, giving them twice the computing power to work with. Still, despite aggressive optimization, training the network took five to six days with the hardware available in 2012 (a pair of Nvidia GTX 580 GPUs, each with 3GB of memory).



(Python) software

University of Ljubljana
Faculty of Mathematics and Physics

- **sklearn (Scikit-Learn)**: Machine learning python open source package.
- **TensorFlow**: Google Deep Neural Network (open source, GPU) , **Keras** (popularni vmesnik - front end).
- **CatBoost**: Popularni vmesnik za pospešena odločitvena drevesa (BDT)
- **Pytorch**: Machine learning python open source package (by Facebook).
- **caffe, theano, minerva**: Other (deep) neural network software.
- **matplotlib, numpy, scipy**: basic sci. libs, plotting and visualization
- **opencv**: Computer vision.
- **pandas**: Data analysis.
- **spyder**: The front end (Scientific PYthon Development EnviRonment).
- Jupyter notebooks etc..
- **Obstajajo tudi verzije za druge jezike, npr CERN-ov TMVA v okolju ROOT..**



Reprezentativni vzorec:Iris

Meet the data: The Iris Dataset

No. of training examples(instances):150

Number of features (x's) : 4

Number of classes : 3

Attribute Information:
Features :

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm

Classes:
-- Iris Setosa
-- Iris Versicolour
-- Iris Virginica

Reprezentativni vzorec: MNIST

- Na roko napisane številke 0-9:
  - Velikost vzorca: 70000 vnosov.
  - Slikice 28x28 točk.

Reprezentativni vzorci: ImageNET

University of Ljubljana
Faculty of Mathematics and Physics

↳... pač vedno večji in večji, celoten spisek na Wikipediji ...

IMAGENET

14,197,122 images, 21841 synsets indexed

Explore Download Challenges Publications Updates About

Not logged in. Login | Signup

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.
Click here to learn more about ImageNet, Click here to join the ImageNet mailing list.

What do these images have in common? *Find out!*

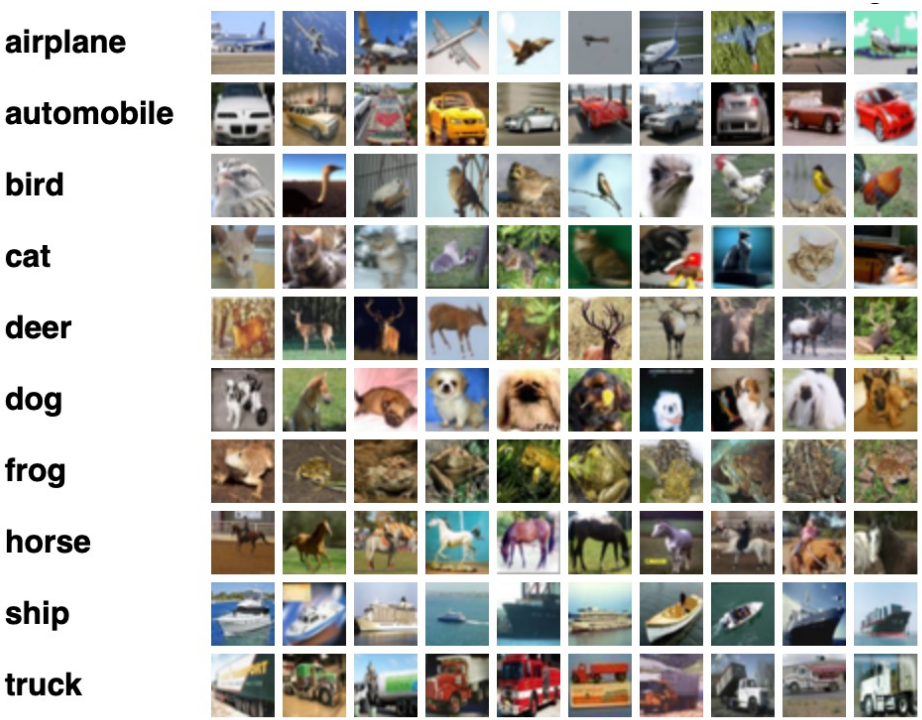
Reprezentativni vzorec: CIFAR-10 in 100

University of Ljubljana
Faculty of Mathematics and Physics

↳ CIFAR-10 dataset (Canadian Institute For Advanced Research):

↳60000 32x32 colour images in 10 classes, with 6000 images per class.

↳There are 50000 training images and 10000 test images.



Reprezentativni vzorec: Higgs Data Set

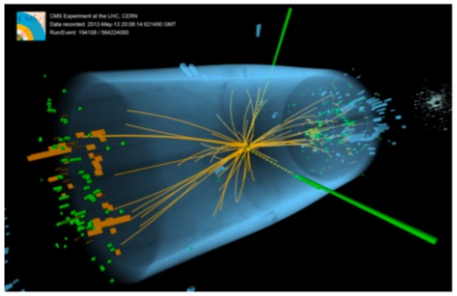
University of Ljubljana
Faculty of Mathematics and Physics

↳ 11 M simuliranih dogodkov, mešanica signala (Higgs) in ozadja (vse ostalo) - narejeni v sodelovanju s CERN-om.

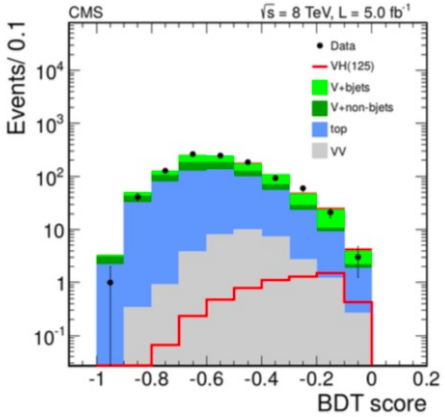
↳ Vsak dogodek ima 28 rekonstruiranih (fizikalnih) količin: g.k., E., inv. M ...

↳ Eden od 'benchmark' vzorcev za določanje uspešnosti algoritmov kategorizacije.

UCI Machine Learning Repository: HIGGS Data Set



BDT



HIGGS Data Set

Download: Data Folder, Data Set Description

Abstract: This is a classification problem to distinguish between a signal process which produces Higgs bosons and a background process which does not.

Data Set Characteristics:	N/A	**Number of Instances:**	11000000	**Area:**	Physical
Attribute Characteristics:	Real	**Number of Attributes:**	28	**Date Donated**	2014-02-12
Associated Tasks:	Classification	**Missing Values?**	N/A	**Number of Web Hits:**	93059

Reprezentativni vzorec: Higgs Data Set

University of Ljubljana
Faculty of Mathematics and Physics

↳ Uporaba DNN za študijo uspešnosti ML za iskanje signala Higgsovega bozona v članku revije Nature Communications:

nature COMMUNICATIONS

ARTICLE

Received 19 Feb 2014 | Accepted 4 Jun 2014 | Published 2 Jul 2014

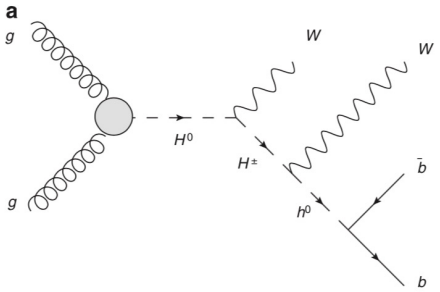
DOI: 10.1038/ncomms5308

Searching for exotic particles in high-energy physics with deep learning

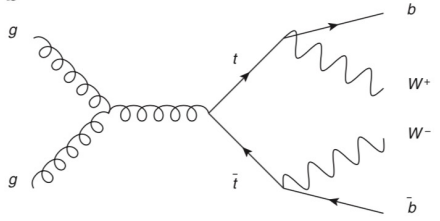
P. Baldi[1], P. Sadowski[1] & D. Whiteson[2]

Zelo podobno končno stanje! (signal je črn, ozadje rdeče)

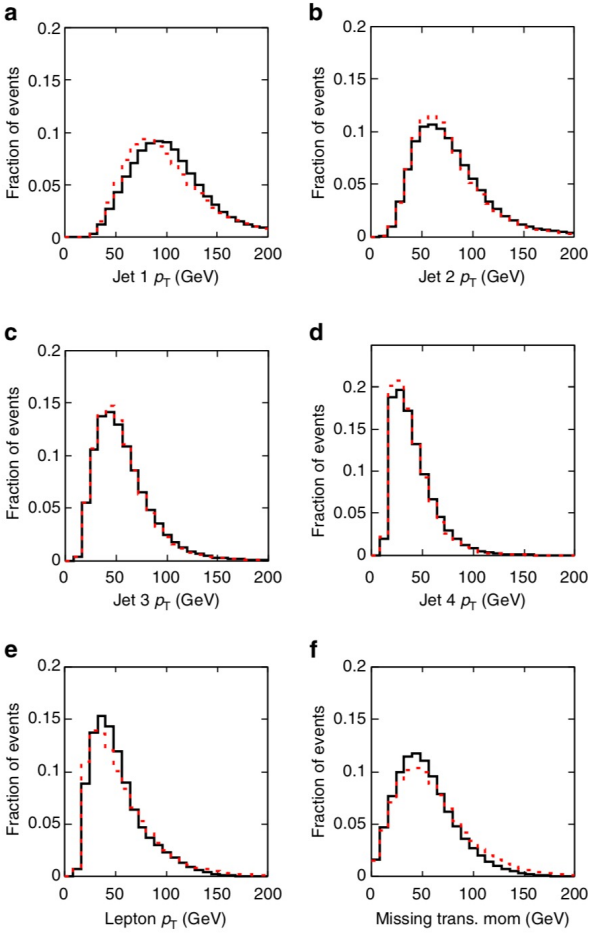
Higgs (signal)



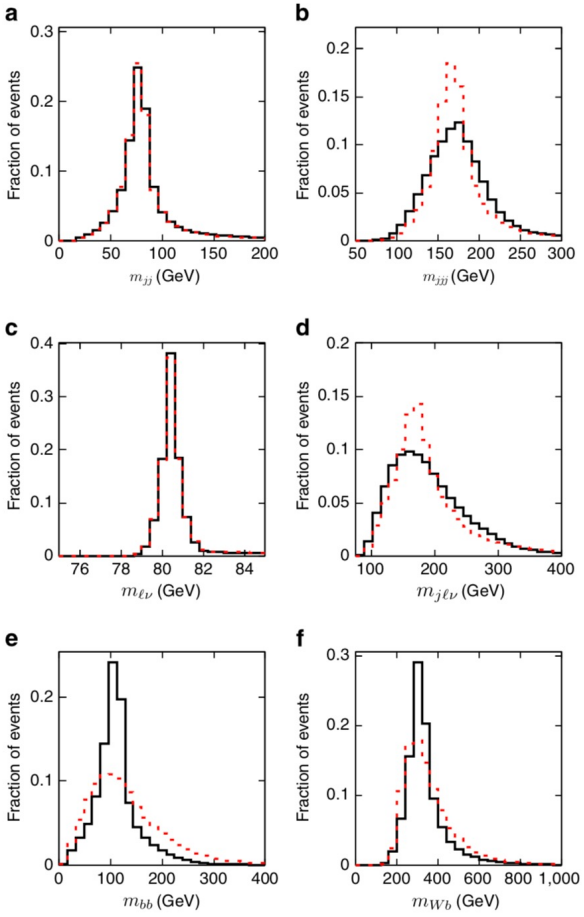
top kvarki (ozadje)



low-level quantities



high-level quantities



Definicija in klasifikacija ML

University of Ljubljana
Faculty of Mathematics and Physics

Definition:

—Tom Mitchell, *1997*

A computer program is said to 'learn' from experience E with respect to some class of tasks *T* and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E

↳*Še prevod: ML program se 'uči' iz izkušnje **E**, glede na neko kategorijo nalog **T** in mero uspeha **P** takrat, ko se pri opravljanju nalog **T** uspeh, merjen z **P**, izboljšuje z izkušnjami **E**.*

↳*naloga **T**: prepoznaj ustrezne signale, napovej naslednji rezultat.*

↳*mera uspeha **P**: odstopanje od pravega rezultata (delež, $\chi 2$, entropija, loss function ...).*

↳*Izkušnje **E**: podatki (dandanes 'big data', miljarde meritev/dogodkov/vnosov)*

Vrste strojnega učenja

University of Ljubljana
Faculty of Mathematics and Physics

↳ Poznamo tri osnovne vrste:

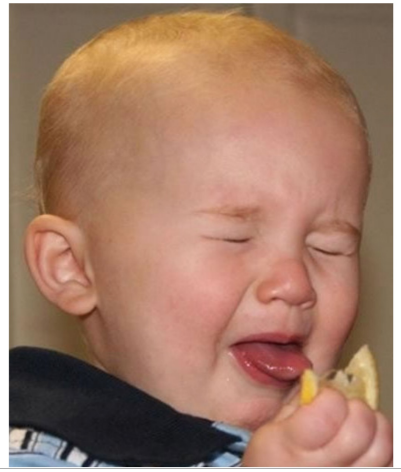
- ↳ **Nadzorovano učenje** (Supervised learning):
  - ↳ **Klasifikacija** (Classification): sortiranje v različne kategorije.
  - ↳ **Regresija** (Regression): fitanje, napovedi.
- ↳ **Nenadzorovano učenje (npr. sam najdi kategorije).**
- ↳ **Stimulirano učenje (Artificial Intelligence v ožjem pomenu besede).**

Danes ...

Supervised learning

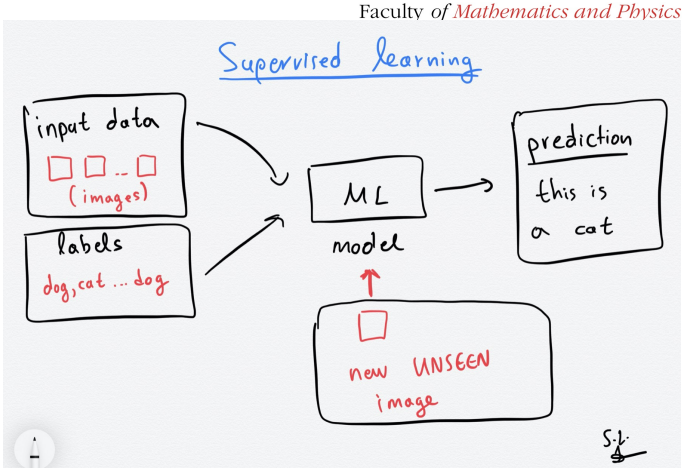
Unsupervised learning

Reinforcement learning



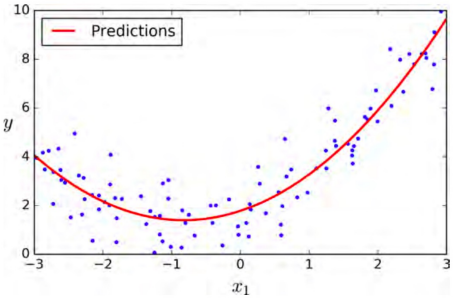
Nadzorovano učenje

- Podatki, na katerih se uči ML algoritem vsebujejo:
  - nabor lastnosti oz karakteristik **(X, features data)** in
  - pravilen odgovor **(Y, label)** za vsak vnos.
- Algoritem mora **generalizirati napoved**, da ustrezno označi tudi nove podatke, ki jih še ni videl.
  - Skratka, **učenje na primerih.**

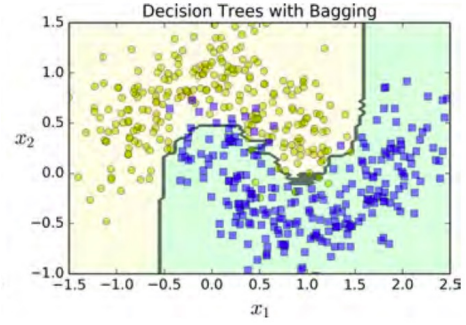


- **Klasifikacija:** razdelitev podatkov v diskretne kategorije:
  - Najdi meje med lastnostmi (**decision boundaries**), ki ločijo med različnimi odgovori/oznakami (labels).
    - Pri klasifikaciji lahko tipično določimo tudi **številsko napoved za izbiro ('verjetnost' , delež dosedanjih uspehov ipd..).**

- **Regresija:** predpostavi, da so lastnosti in odgovor povezani z neko matematično funkcijo (**Y = h(X)**) in torej v resnici iščeš to funkcijo - torej gre za iskanje neznane funkcije s približki oziroma interpolacijo.



- **ML algoritmi imajo prednost pred klasičnim pristopom, da lahko učinkovito razdrobijo kompleksen problem na enostavne elemente in ga ustrezno opišejo:**
  - pomisli na primer, kako bi bilo težko kar predpostaviti/uganiti pravo analitično funkcijo v več dimenzijah (in je npr. spline interpolacija mnogo lažja in boljša).
  - Pri izbiri/filtriranju velike količine podatkov z mnogo lastnostmi (npr dogodki pri trkih na LHC, zvezdnih kopicah..) je zelo težko najti količine, ki optimalno ločijo signal od ozadnja, upoštevati vse korelacije in najti optimalno kombinacijo le-teh...



Nenadzorovano učenje

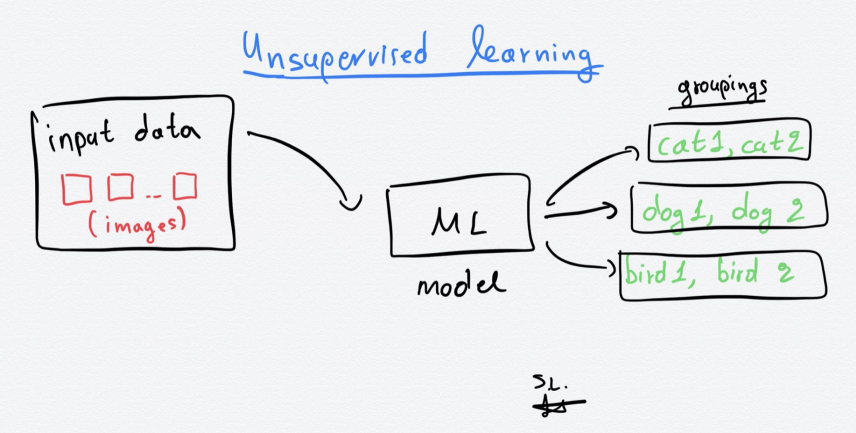
University of Ljubljana
Faculty of Mathematics and Physics

↳ Unsupervised learning:

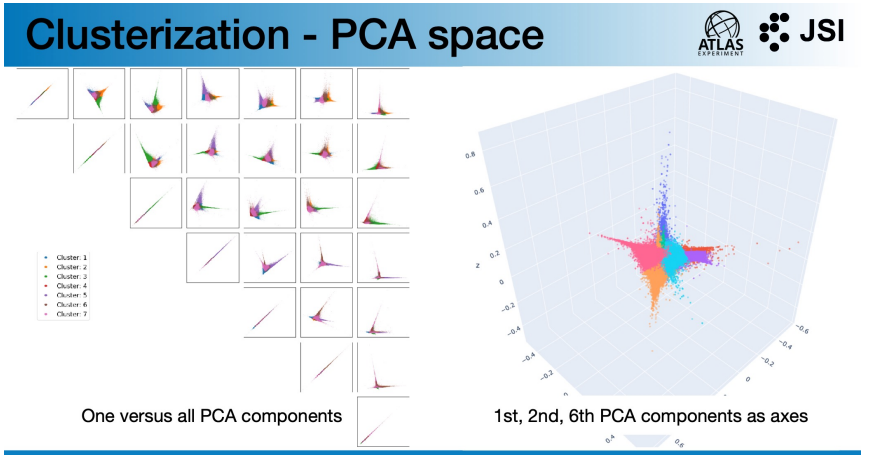
↳ Najboljši zgled in danes glavna raba je združevanje v gruče (clustering):

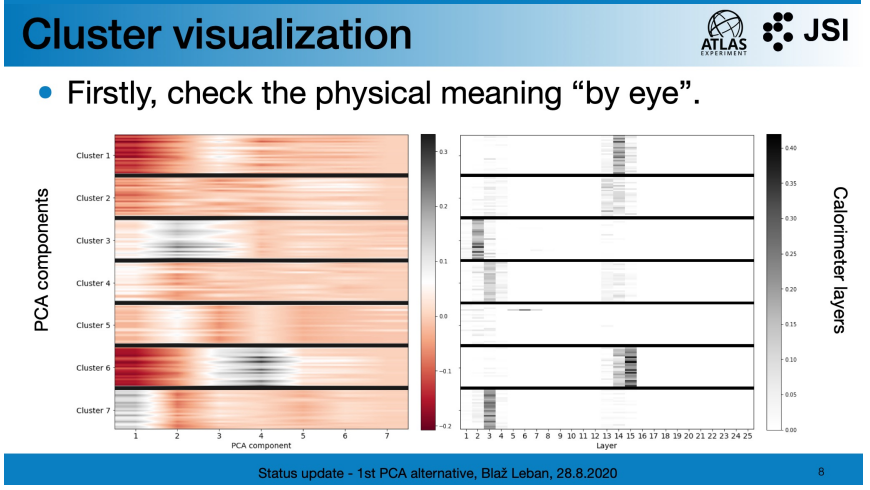
↳ V množici podatkov algoritem sam glede na lastno metriko združi podatke v skupine (brez nadzora).

↳ Rezultat je pogosto presenetljiv, a včasih uporaben ...



sample Cluster/group





Definicija uspešnosti: Klasifikacija

University of Ljubljana
Faculty of Mathematics and Physics

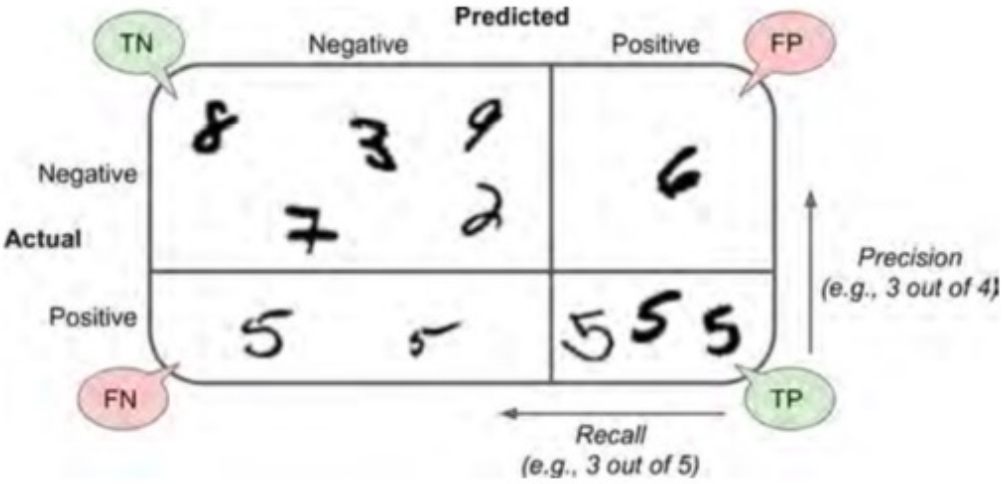
↳Več možnosti, tipično napaka prve ali druge vrste med dvema hipotezama v kombinaciji z uspešnostjo (izkoristek, efficiency) izbire prave hipoteze:

TN = True Negative = correctly rejected bkg

Confusion matrix = $\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$

$$\text{recall (efficiency, sensitivity)} = \frac{TP}{TP+FN}$$

$$\text{precision (purity,effectivity)} = \frac{TP}{TP+FP}$$

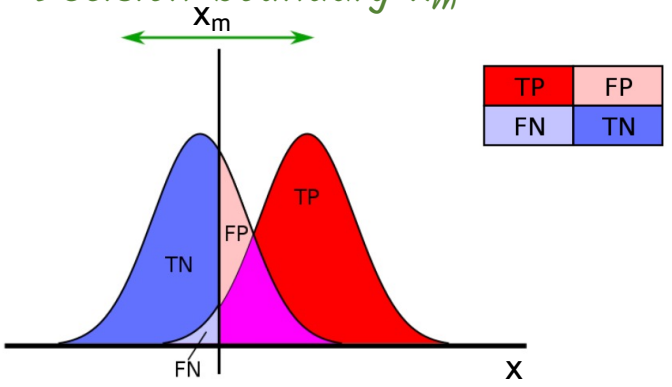


Statistično testiranje hipotez (Wikipedia):

↳**napaka 1. tipa**: zavrgli smo pravilen dogodek.
Klasično: rate α: significance, α-level,
ML: False Negative rate (FN),
Fizikalno: $1-\varepsilon$, ε=izkoristek (efficiency)

↳**napaka 2. tipa**: sprejeli smo napačen dogodek
Klasično: rate β: power of a test=$(1-\beta)$ level
ML: False Positive rate (FP),
Fizikalno: kontaminacija, čistost (purity)

Decision boundary x_m



Definicija uspešnosti: Klasifikacija

University of Ljubljana
Faculty of Mathematics and Physics

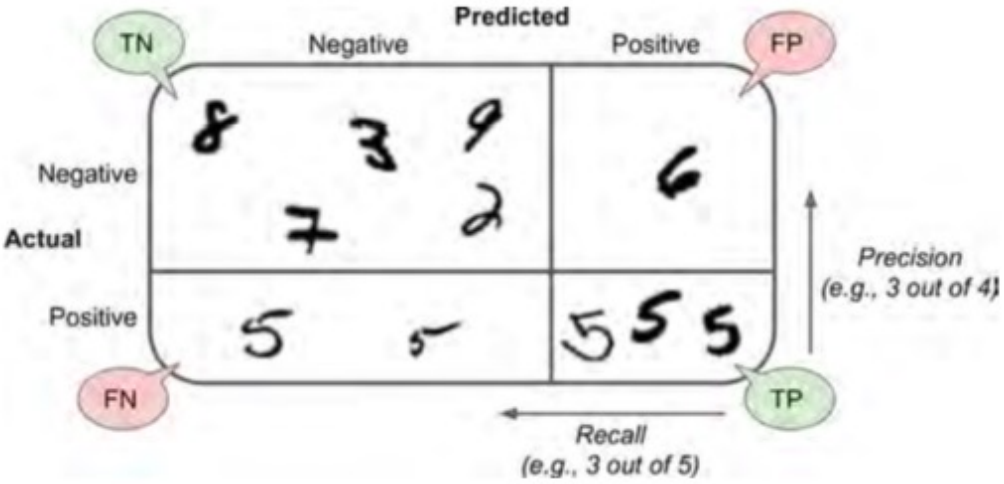
↳Več možnosti, tipično napaka prve ali druge vrste med dvema hipotezama v kombinaciji z uspešnostjo (izkoristek, efficiency) izbire prave hipoteze:

TN = True Negative = correctly rejected bkg

Confusion matrix = $\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$

$$\text{recall (efficiency, sensitivity)} = \frac{TP}{TP+FN}$$

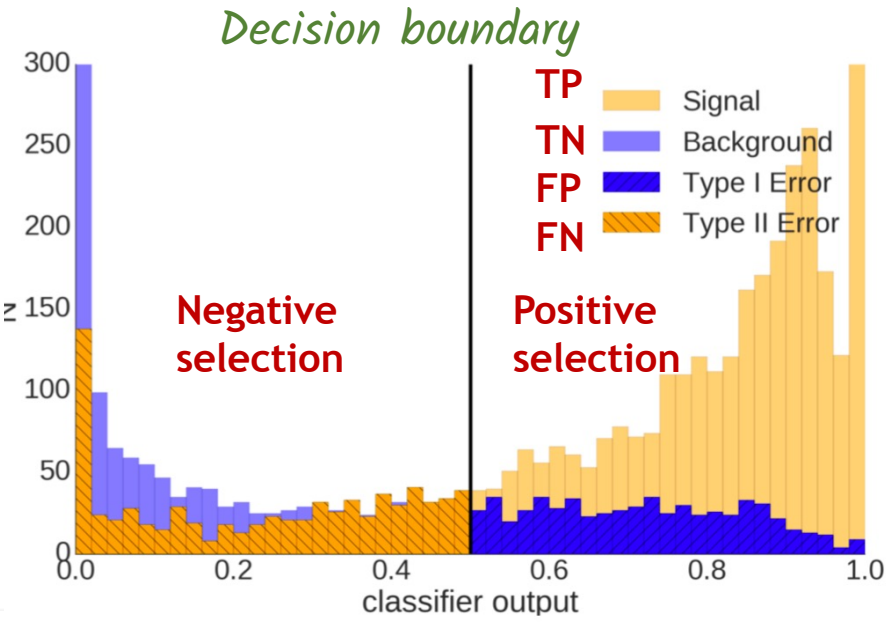
$$\text{precision (purity,effectivity)} = \frac{TP}{TP+FP}$$



Statistično testiranje hipotez (Wikipedia):

↳**napaka 1. tipa**: zavrgli smo pravilen dogodek.
Klasično: rate α: significance, α-level,
ML: False Negative rate (FN),
Fizikalno: $1-\varepsilon$, ε=izkoristek (efficiency)

↳**napaka 2. tipa**: sprejeli smo napačen dogodek
Klasično: rate β: power of a test=$(1-\beta)$ level
ML: False Positive rate (FP),
Fizikalno: kontaminacija, čistost (purity)



Definicija uspešnosti: Klasifikacija

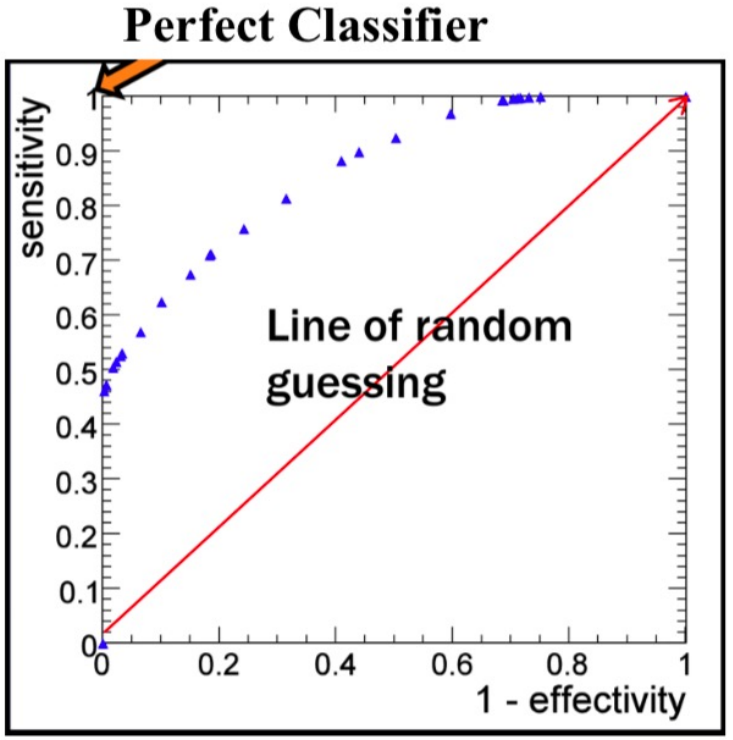
University of Ljubljana
Faculty of Mathematics and Physics

Receiver Operating Characteristic (ROC)

Commonly used metric

Shows the **relationship** between correctly classified positive cases (sensitivity) and incorrectly classified negative cases (1-effectivity)

Function of decision boundary or equivalent ...



False positive vs true positive rate (efficiency vs 1-purity)

Definicija uspešnosti ML

↳ Za uspeh ML definiramo funkcijo izgube/uspeha (**loss ali cost function J**) najpogosteje kar kot ekvivalent $\chi 2$ oz Mean Square Error (MSE) (ali RMSE ali..) glede na iskano funkcijo y=h(X):

$$\mathrm{MSE}(\mathbf{X}, h) = \frac{1}{m_i}\sum_{i=1}^{m}\left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2 \qquad \mathrm{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2} \qquad \mathrm{MAE}(\mathbf{X}, h) = \frac{1}{m}\sum_{i=1}^{m}\left|h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right|$$

↳ Posebej zanimiv zaradi povezave z nevronskimi mrežami je tudi **Logistic Regression**: določi oceno za verjetnost **p** glede na vhodne lastnosti **X pri y = 0 ali 1** in modelske parametre ϑ (**v bistvu verjetnost za kategorizacijo**):

$$\hat{p} = h_\theta(\mathbf{X}) = \sigma\left(\theta^T \cdot \mathbf{X}\right)$$

linearna funkcija: $b+\vartheta_1 x_1+\vartheta_2 x_2+....+\vartheta_N x_N$

$$\sigma(t) = \frac{1}{1+e^{-t}}$$

Logistic function

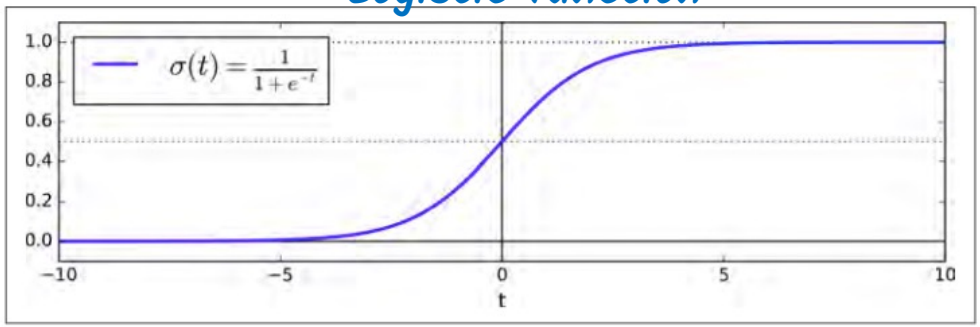


Figure 4-21. Logistic function

Cost function (y = 0 ali 1 - true/false)

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(\hat{p}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-\hat{p}^{(i)}\right)\right]$$

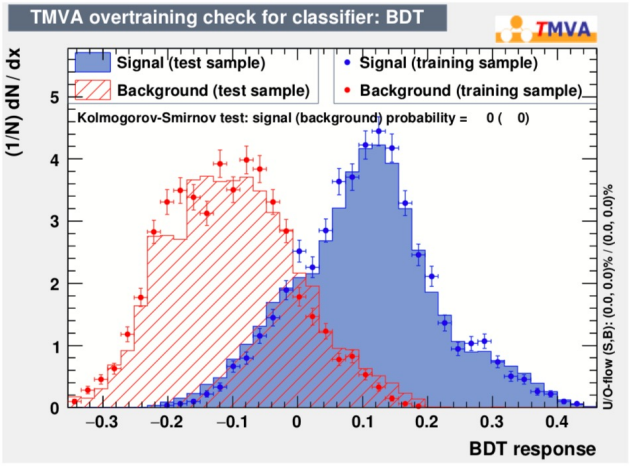
Njen odvod — za Newtonovo (gradient) metodo

$$\frac{\partial}{\partial\theta_j}\mathrm{J}(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\sigma\left(\theta^T \cdot \mathbf{X}^{(i)}\right) - y^{(i)}\right)X_j^{(i)}$$

Priprava vzorcev podatkov in učenje

↳**Pri pripravi ML algoritma potrebuješ podatke za dva ločena koraka:**

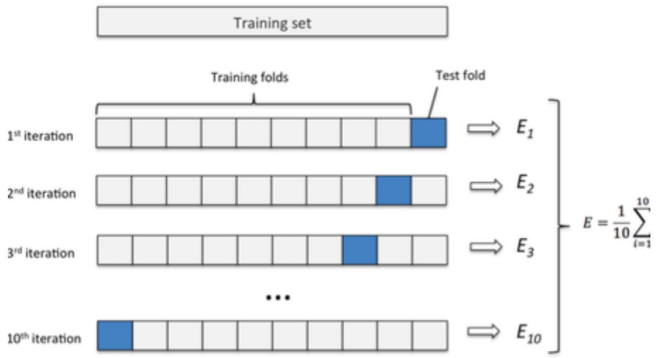
- ↳1) korak učenja ML algoritma,
- ↳2) korak validacije rezultatov.
- ↳**Pri obeh korakih uporabi različne vzorce!**
- ↳... In ponavljaj in ponavljaj do želenega uspeha.
- ↳Naprednejši pristop je **cross-validation**, kjer recikliramo cel vzorec za vsako ponovitev/iteracijo.
- ↳**Ne pozabi na neodvisni vzorec za končni test (na LHC so to merjeni podatki).**

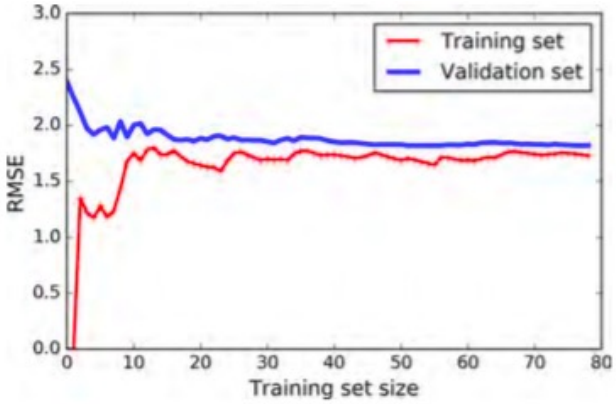


↳**Bistvenega pomena je tudi 'normalizacija' lastnosti vzorcev!**

- ↳Praktično vsi algoritmi slabo regirajo na lastnosti z razponi v različnih velikostnih redih - torej, **najdi brezdim. količino in/ali reskaliraj:**
  - ↳**min-max pristop**: reskaliraj vse lastnosti v isti interval (npr [0.,1.])
  - ↳**Normalizacija v smislu Gaussove porazdelitve** (z = (x - AVG(x)) /RMS(x)).

Razmisli o vhodnih spremenljivkah, parametrizaciji problema ipd! GIGO....





Metode ML učenja v dejanski izvedbi

↳Princip učenja v ML je ekvivalenten logiki v fitanju, torej minimizaciji 'cost' oz. 'loss' funkcije (Mean Square Error (MSE), Entropijske funkcije...) pri določanju parametrov v ML mehanizmu.

↳Kot pri minimizaciji, iščemo minimum (torej ničle v odvodu); tehnično ime (tipičnega) osnovnega postopka je **Gradient Descent, kar ni nič drugega kot Newton-Rhapson metoda! (z vsemi znanimi prednostmi in težavami).**

↳Seveda potem izvedenke, kot so Stochastic Gradient Descent, momentum back-propagation , ADAM algoritem v DN ipd..

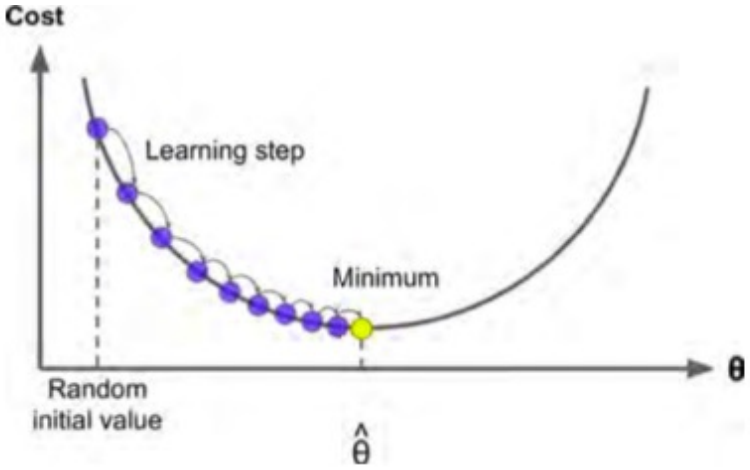
↳**TensorFlow ima posebej vgrajeno tudi funkcijo za izračun jakobijevih matrik!**

$$\theta^{(\text{next step})} = \theta - \eta \nabla_\theta J(\theta)$$

Equation 11-8. Adam algorithm

1. $\mathbf{m} \leftarrow \beta_1 \mathbf{m} + (1 - \beta_1) \nabla_\theta J(\theta)$
2. $\mathbf{s} \leftarrow \beta_2 \mathbf{s} + (1 - \beta_2) \nabla_\theta J(\theta) \otimes \nabla_\theta J(\theta)$
3. $\mathbf{m} \leftarrow \dfrac{\mathbf{m}}{1 - \beta_1^{\,T}}$
4. $\mathbf{s} \leftarrow \dfrac{\mathbf{s}}{1 - \beta_2^{\,T}}$
5. $\theta \leftarrow \theta - \eta\, \mathbf{m} \oslash \sqrt{\mathbf{s} + \epsilon}$

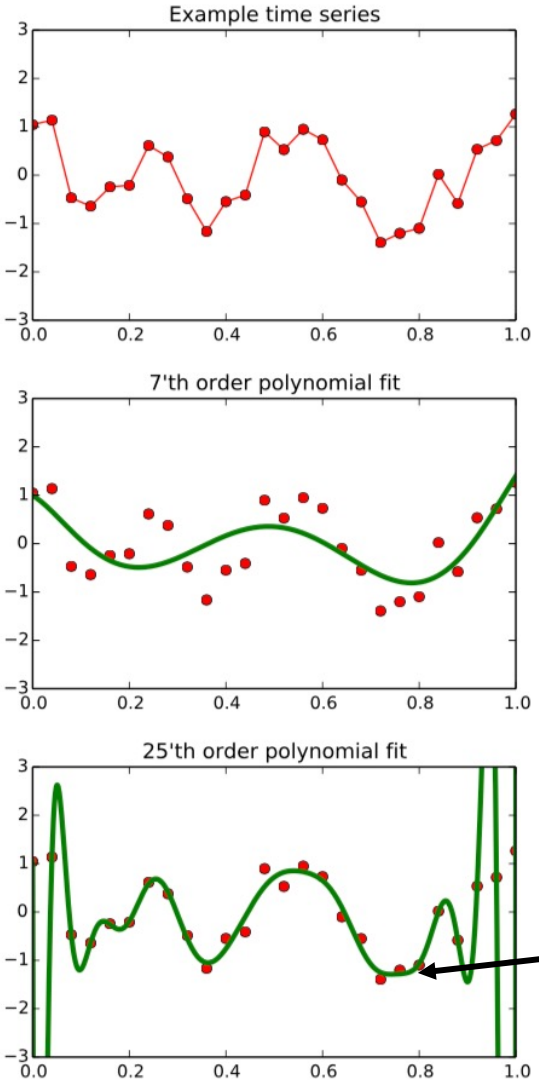
- T represents the iteration number (starting at 1).

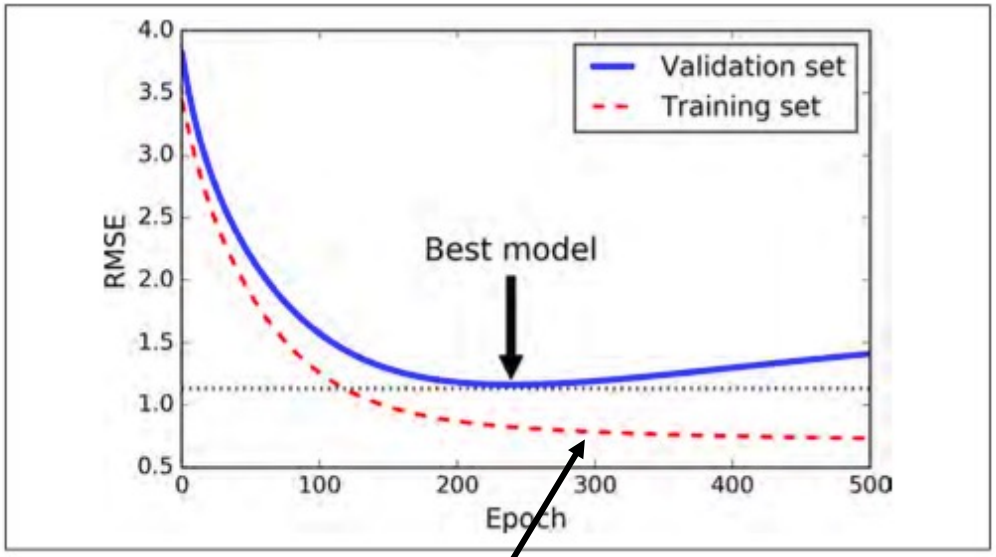


Pazi: Training in Over-training

↳ Kdaj končati z učenjem? **Early stopping** je odličen pristop

Analogija z fitanjem/regresijo ...





Over-training!

Over-fitting!

Različni ML algoritmi

University of Ljubljana
Faculty of Mathematics and Physics

↳**V resnici je ML algoritmov mnogo:**

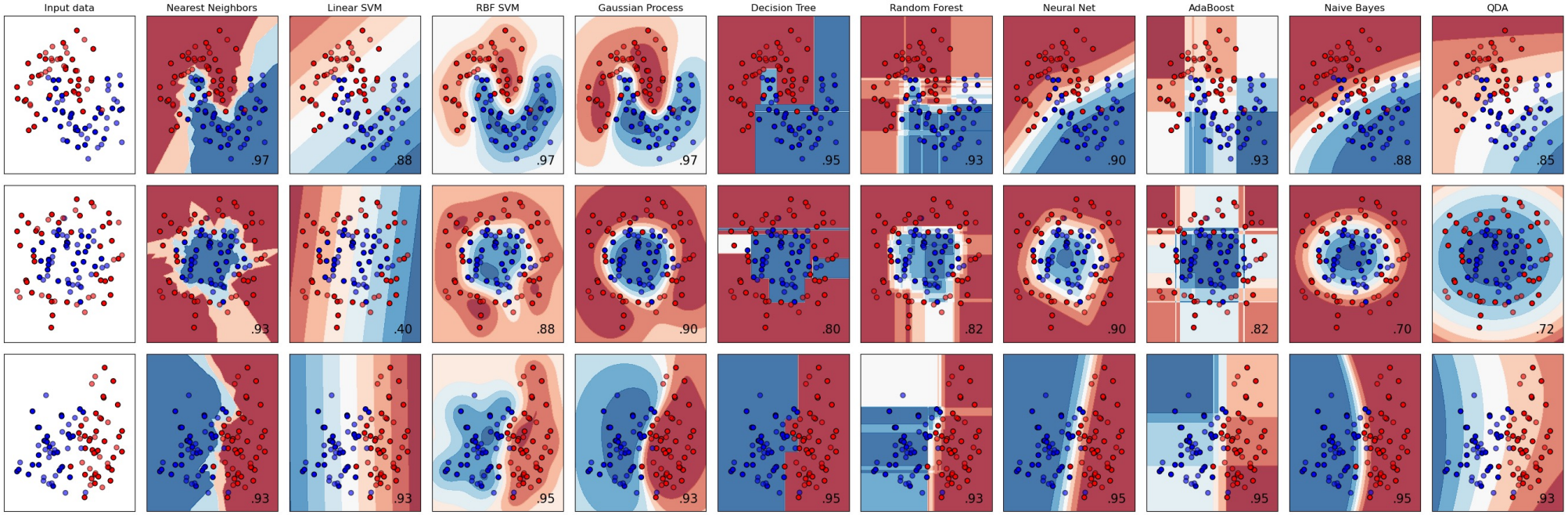
↳**(Artificial) Neural Networks - nevronske mreže** - samo eden od njih.

↳Ogledali si bomo še primer **pospešenih odločitvenih dreves - Boosted Decision Trees (BDT)** kot dober in preprost zgled.

↳Ostalo si poglejte sami (samo pokazal bom zgled)

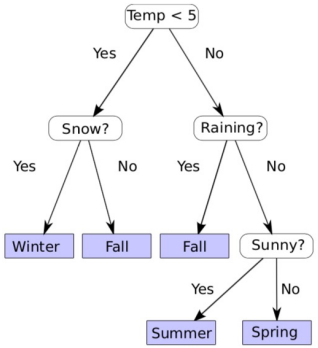
↳Priporočam tudi Support Vector Machines (SVM), ...

↳Samo nekaj implementiranih metod v v Scikit-Learn:

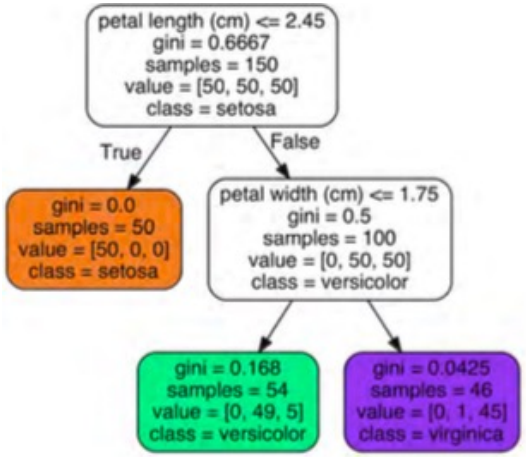


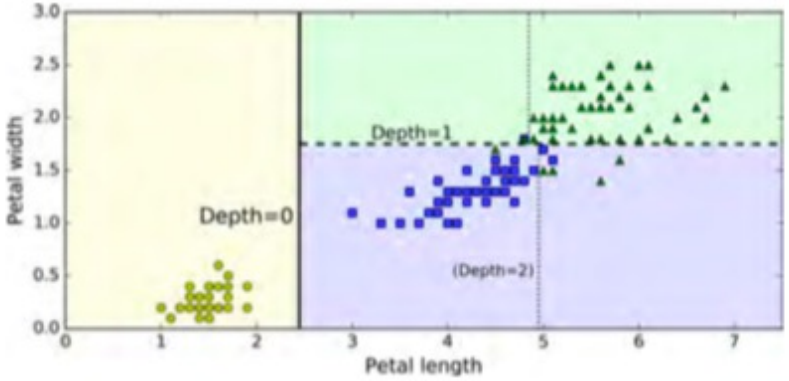
BDT in Random Forests

- **Decision trees** (odločitvena drevesa) so v osnovi zelo preprost sistem odločanja **dalne (if-then-else)** z delitvijo kategorij na pod-kategorije.
  - Uporabna za **klasifikacijo** (seveda) in tudi **regresijo** (stevilčne napovedi)
  - Išče najčistejšo razdelitev vzorcev (klasifikacija) ali minimizira MSE (regresija)

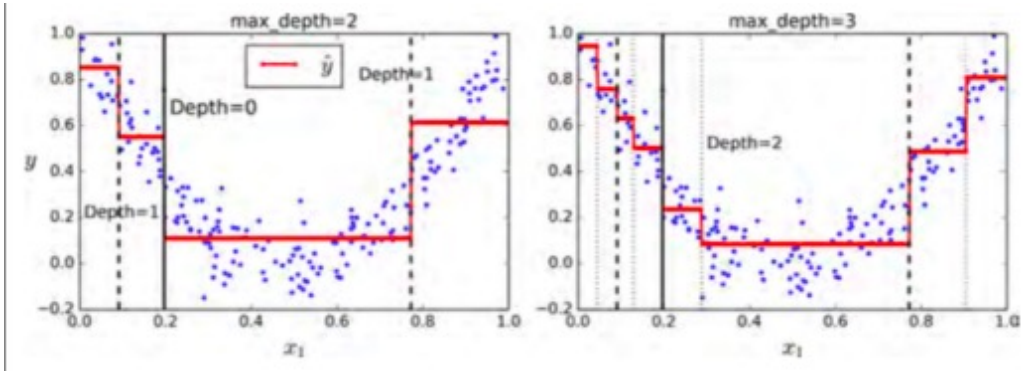


Kategorizacija irisov

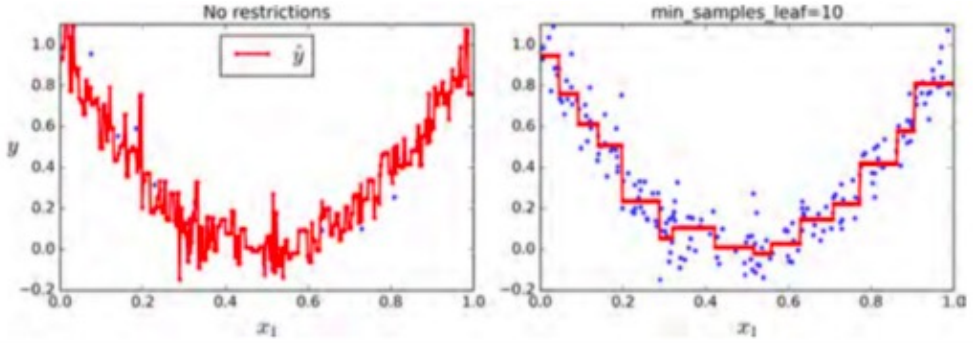




Regresija (fit parabole s kategorijami/predali!)



Regularizacija DT



BDT in Random Forests

↳Sama po sebi DT niso močno orodje/cenilec (**weak estimator**).

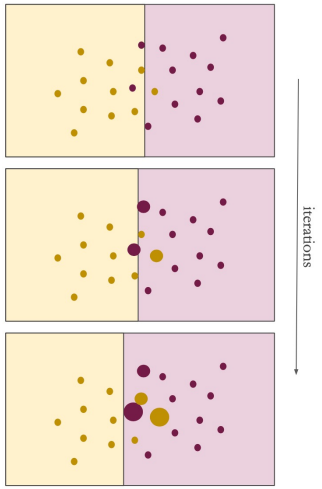
↳problem 'overtraininga', sploh brez omejitev (regularizacije).

↳so pa odlična osnova za **kolektivne (ensemble) metode**:

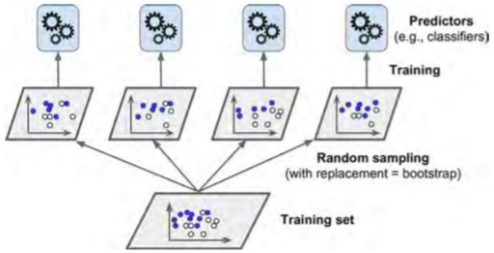
↳**Random Forests**: 'bagging',naključno izbrana drevesa, superpozicija.

↳**Boosted Decision Trees (AdaBoost, Gradient Boost algoritem)**: iterativna metoda, kjer utežimo problematične 'outlierje' .

↳**V obeh primerih je rezultat kombinacija rezultatov vseh dreves.**

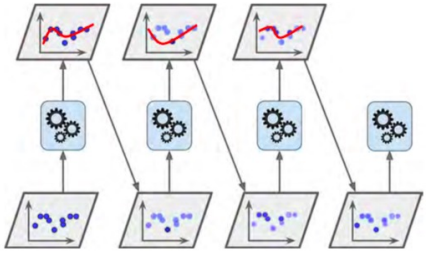


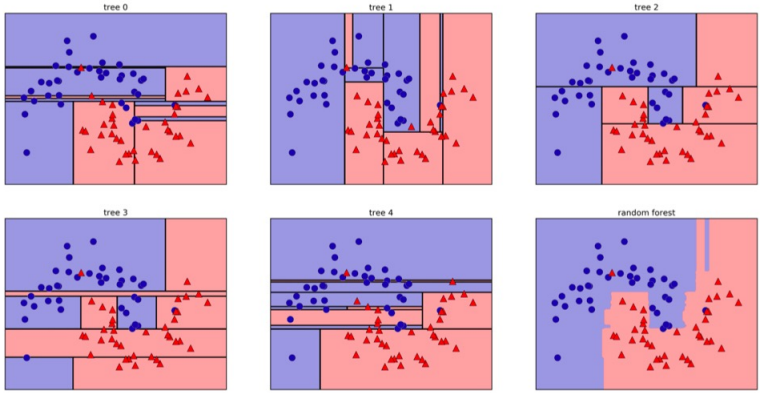
Random forests pristop



Bagging (boostrap): naključno izbiramo vnose/dogodke iz vzorca, ponavljanje dogodkov je dovoljeno.
Pasting: ponavljanje ni dovoljeno.

AdaBoost - BDT pristop





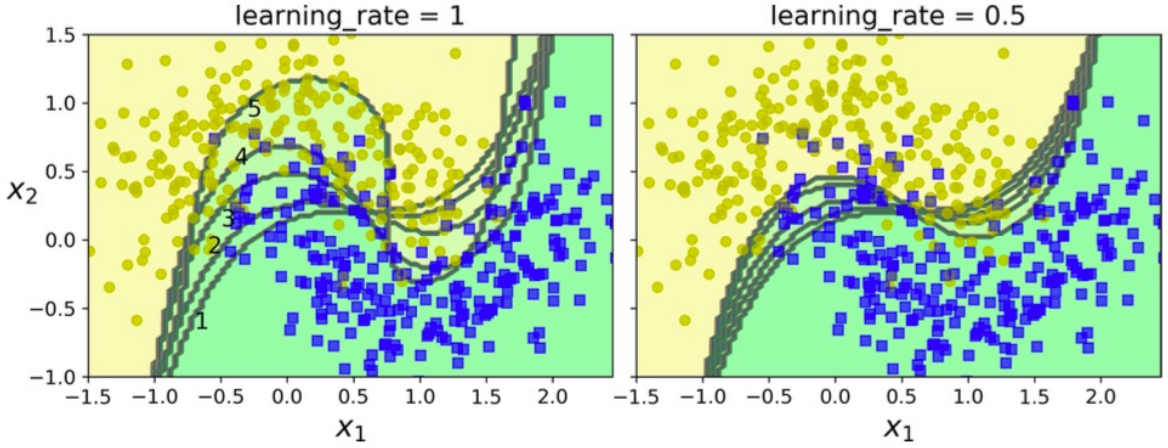
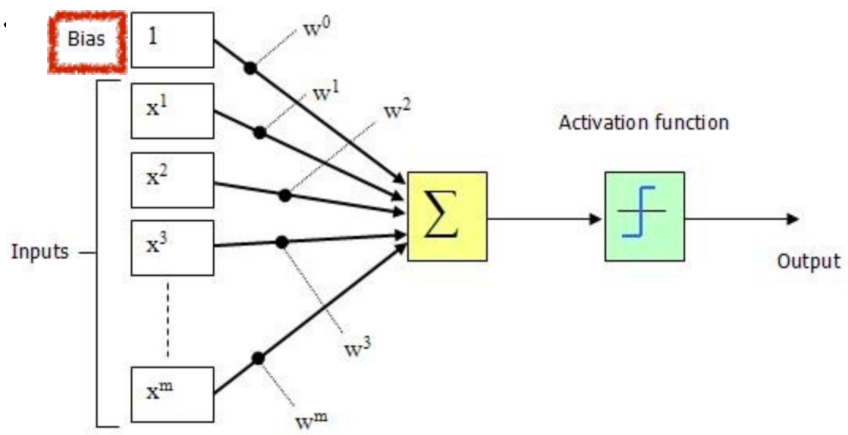


Figure 2-33. Decision boundaries found by five randomized decision trees and the decision boundary obtained by averaging their predicted probabilities

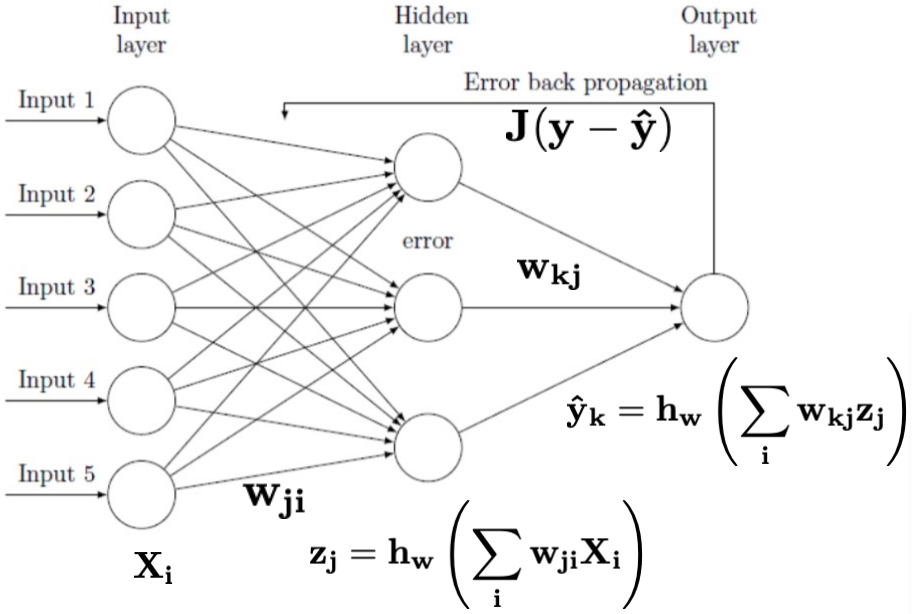
Nevronske mreže

- Umetne nevronske mreže (Artificial Neural Networks, NN):
- **Umetni nevron** (**perceptron**, originalno **1957** Frank Rosenblatt):
  - Aktivacijska funkcija kot odziv na vhodne signale.
- **Povezava** nevronov v **plasti** (layers) in plasti v **mreže**
  - z **različnimi** povezavami med njimi: dense ali convolutional ali ...
- **Uporabljamo prvenstveno za kategorizacijo!**
  - **Učenje**: (back-)propagacija napake napovedi (oz. minimizacija cost funkcije)

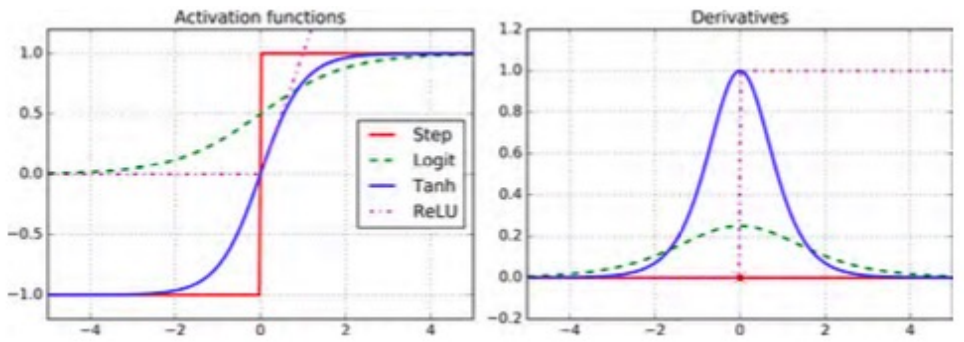
Perceptron: Nevron v (A)NN



Preprosta shema (A)NN



$$h_w(\mathbf{X}) = \text{actfun}\left(\mathbf{w}^T \cdot \mathbf{X}\right)$$



Tipično učimo z (X,y=[0,...1,..0]) , uteži w so naši modelski parametri

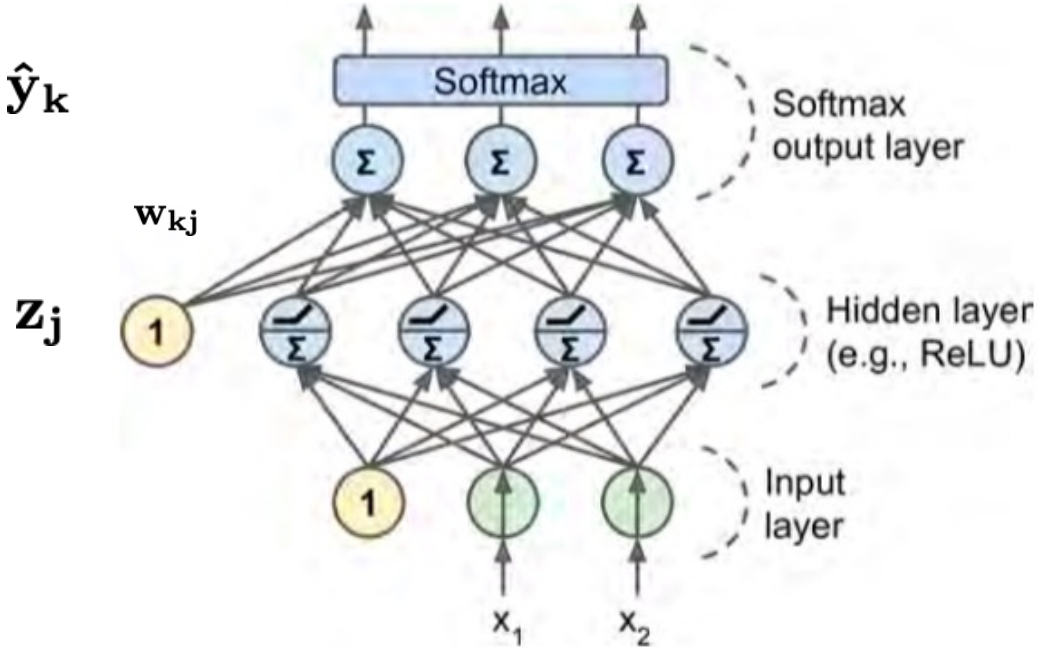
kategorije

Nevronske mreže za kategorizacijo

Tipično učimo z (X,y=[0,..1,..0])

kategorije

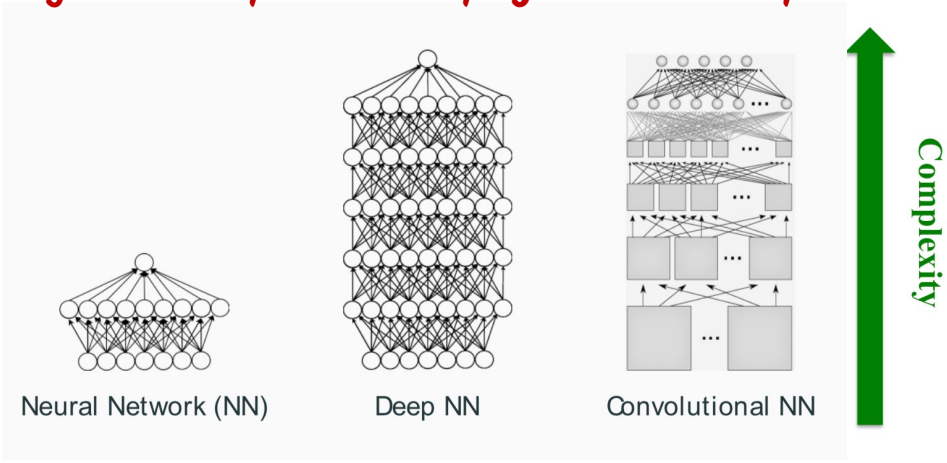
Shematična nevronska mreža



- Bolj natančno: za kategorizacijo tipično softmax (razširjen logistic) odziv na izhodni plasti.
  - Analogno razširjena cost funkcija (t.i. cross-entropy)
  - Naš izhodni vektor $\hat{y}$ ima torej vrednosti izražene kot verjetnosti v intervalu (0,1).

$$\hat{y}_k = \hat{p}_k = \sigma\left(\sum_j w_{kj} z_j\right) = \frac{\exp\left(\sum_j w_{kj} z_j\right)}{\sum_{k=1}^{K} \exp\left(\sum_j w_{kj} z_j\right)}$$

Zgodba se potem še poljubno zakomplicira

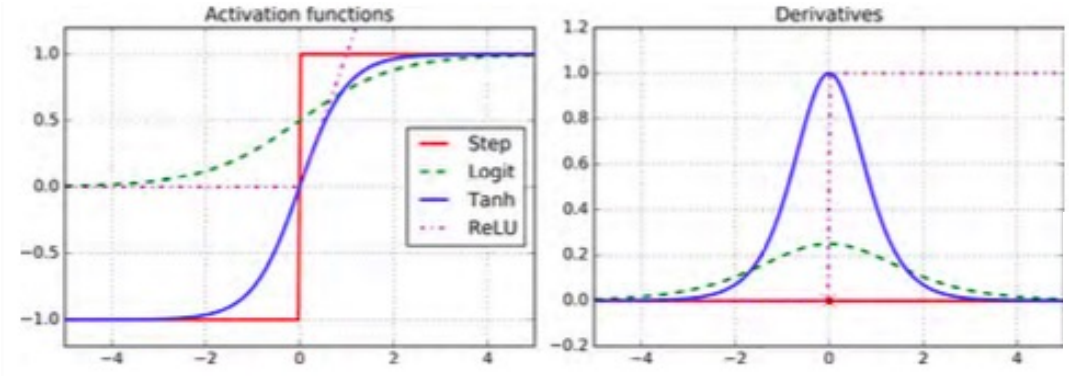


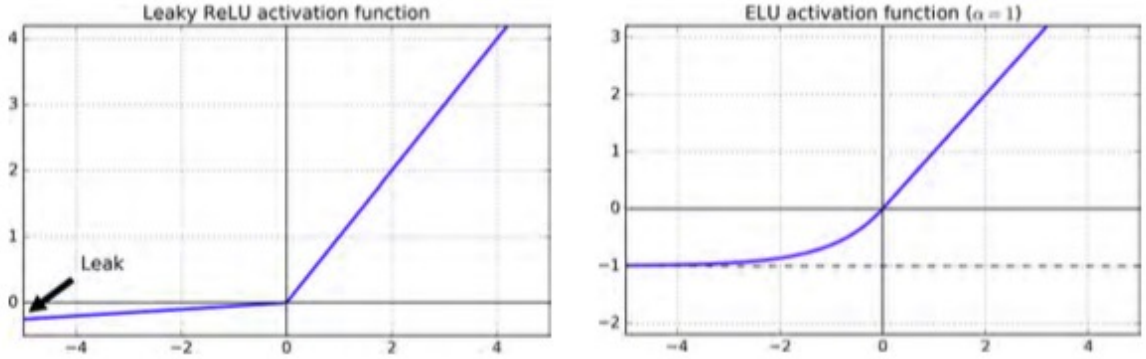
$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} y_k^{(i)} \log\left(\hat{p}_k^{(i)}\right)$$

```
model = tf.keras.Sequential()
# Adds a densely-connected layer with 64 units to the model:
model.add(layers.Dense(64, activation='relu'))
# Add another:
model.add(layers.Dense(64, activation='relu'))
# Add a softmax layer with 10 output units:
model.add(layers.Dense(10, activation='softmax'))
```

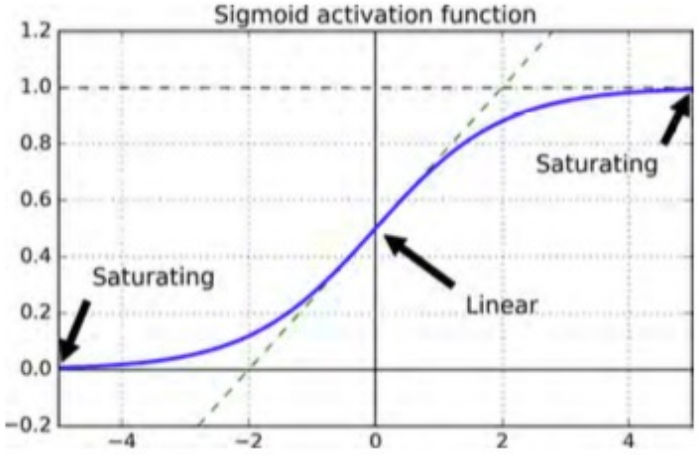
Nevronske mreže: Učenje

- Aktivacijska funkcija: najkompleksnejša ni nujno najboljša!
  - Primer saturacije pri učenju (Newtonova metoda: saturacija pomeni gradient = 0!): **Vanishing gradients problem!**
    - Torej ne znamo napake učinkovito razdeliti nazaj po NN!
    - Najboljši robustna 'Leaky ReLU' in pa recimo ELU





$$\theta^{(\text{next step})} = \theta - \eta \nabla_\theta J(\theta)$$



$$\text{ELU}_\alpha(z) = \begin{cases} \alpha(\exp(z) - 1) & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$$

Regularizacija

University of Ljubljana
Faculty of Mathematics and Physics

↳V vseh metodah ML se uporabi takšna ali drugačna **regularizacija:** omejitev vrednosti in števila parametrov v sistemu, kar pospeši/izboljša generalizacijo problema.

↳**Primer**: pri regresiji (fitanju) so pričakovani parametri fita 'pohlevni', torej dovolj majhni!

↳sploh v fiziki in če smo vhodne podatke ustrezno pripravili/normalizirali!

↳Večina orodij ima te regularizacije že vgrajene!

↳**Preveri za vsako metodo in to uporabi!**

L1 regularizacija (Lasso, Least absolute shrinkage...)

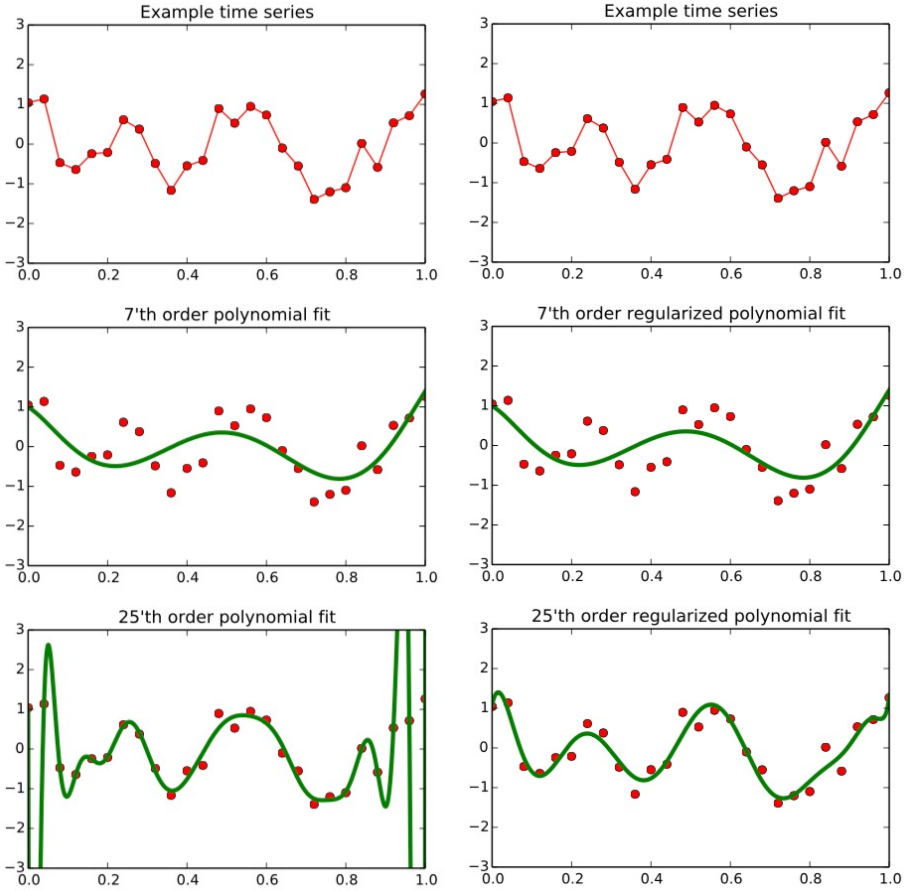
$$J(\theta) = \mathrm{MSE}(\theta) + \alpha \sum_{i=1}^{n} |\theta_i|$$

L2 regularizacija (Tykhonov, Ridge..)

$$J(\theta) = \mathrm{MSE}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$$

Elastic Net regularizacija (kombinacija...)

$$J(\theta) = \mathrm{MSE}(\theta) + r\alpha \sum_{i=1}^{n} |\theta_i| + \frac{1-r}{2} \alpha \sum_{i=1}^{n} \theta_i^2$$



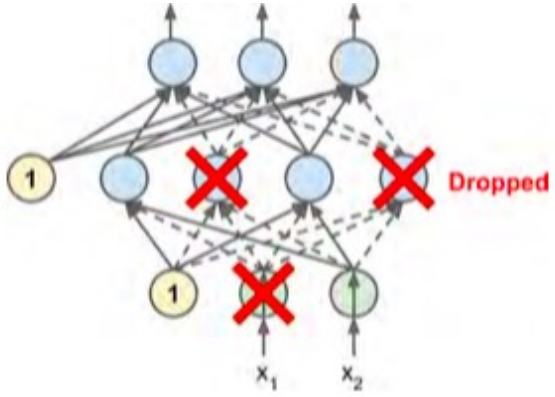
Regularizacija pri ANN

↳V vseh metodah ML se uporabi takšna ali drugačna **regularizacija:** omejitev vrednosti in števila parametrov v sistemu, kar pospeši/izboljša generalizacijo problema.

↳**Primer**: v NN se postopek učenja ponavlja z izključevanjem nevronov (naključno ali sistematično): **Dropout regularization**

↳Zanimivo, da sploh deluje... (zgled 'preveč zaposlenih v podjetju')

↳Tudi L1 in L2 regularizacija na utežeh ...



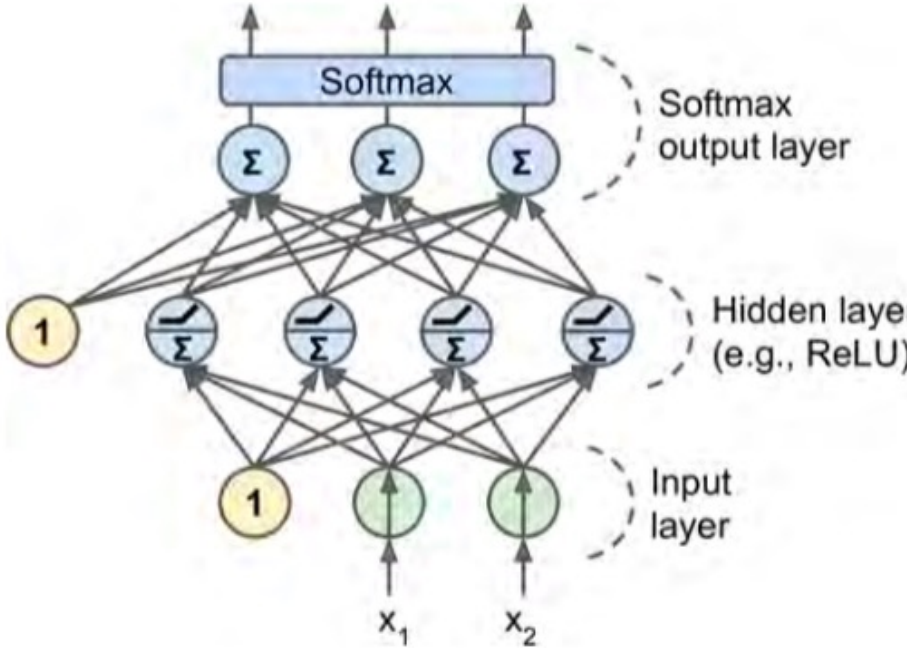

```
# A linear layer with L1 regularization of factor 0.01 applied to the kernel matrix:
layers.Dense(64, kernel_regularizer=tf.keras.regularizers.l1(0.01))

# A linear layer with L2 regularization of factor 0.01 applied to the bias vector:
layers.Dense(64, bias_regularizer=tf.keras.regularizers.l2(0.01))
```

A(NN) Moj doktorat - WW (2000)

University *of Ljubljana*
Faculty *of Mathematics and Physics*

Shematična nevronska mreža



4.5 Reconstruction of $W^{\pm}$ Direction – Pairing of Jets

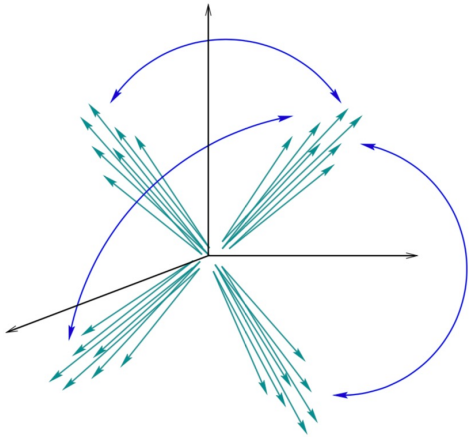
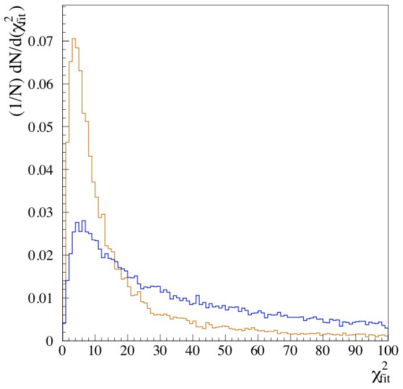
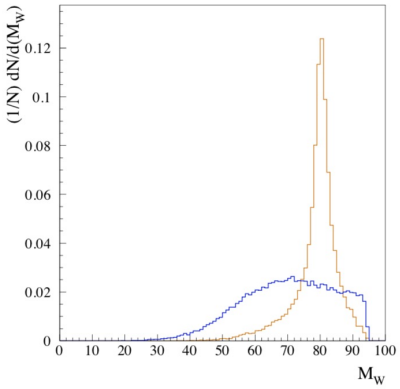
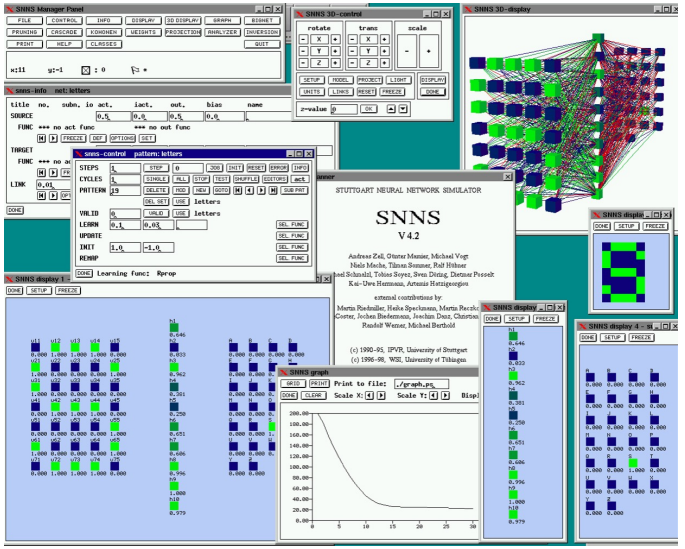


Figure 4.4: A schematic representation of the three possible pairing combinations of jets in a four-jet topology characteristic for the $W^+W^- \to q_1\bar{q}_2q_3\bar{q}_4$ process.







... skratka, upobraba ML v fiziki res ni nova....

Domača naloga: Higgs Data Set

University of Ljubljana
Faculty of Mathematics and Physics

↳ Uporaba DNN za študijo uspešnosti ML za iskanje signala Higgsovega bozona v članku revije Nature Communications:

nature COMMUNICATIONS

ARTICLE

Received 19 Feb 2014 | Accepted 4 Jun 2014 | Published 2 Jul 2014

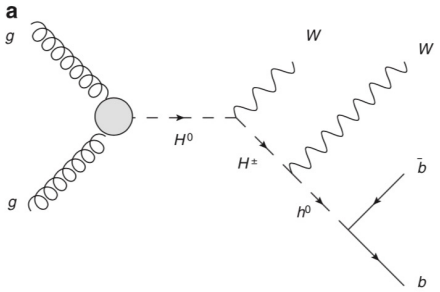
DOI: 10.1038/ncomms5308

Searching for exotic particles in high-energy physics with deep learning

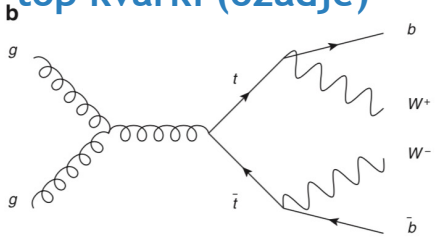
P. Baldi[1], P. Sadowski[1] & D. Whiteson[2]

Zelo podobno končno stanje! (signal je črn, ozadje rdeče)

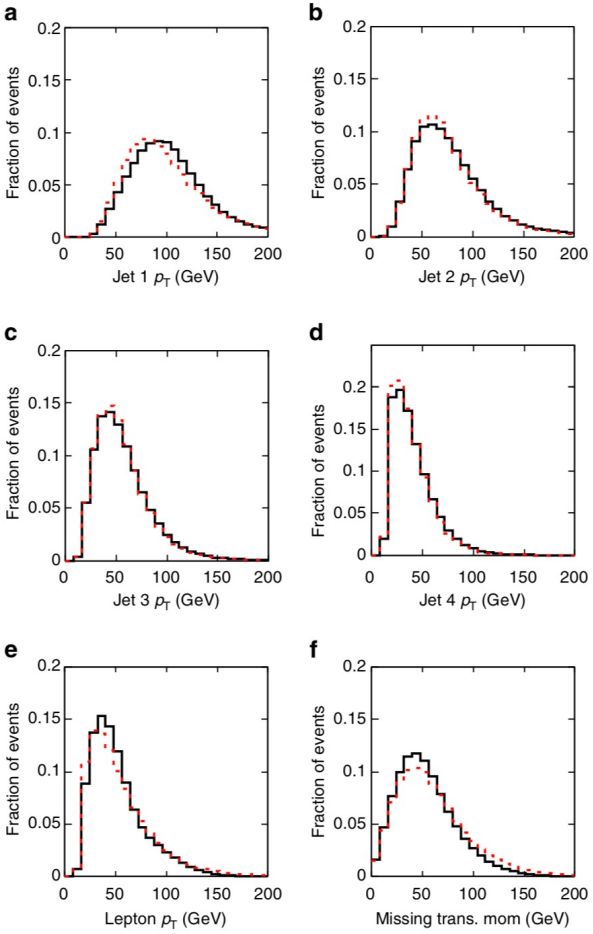
Higgs (signal)



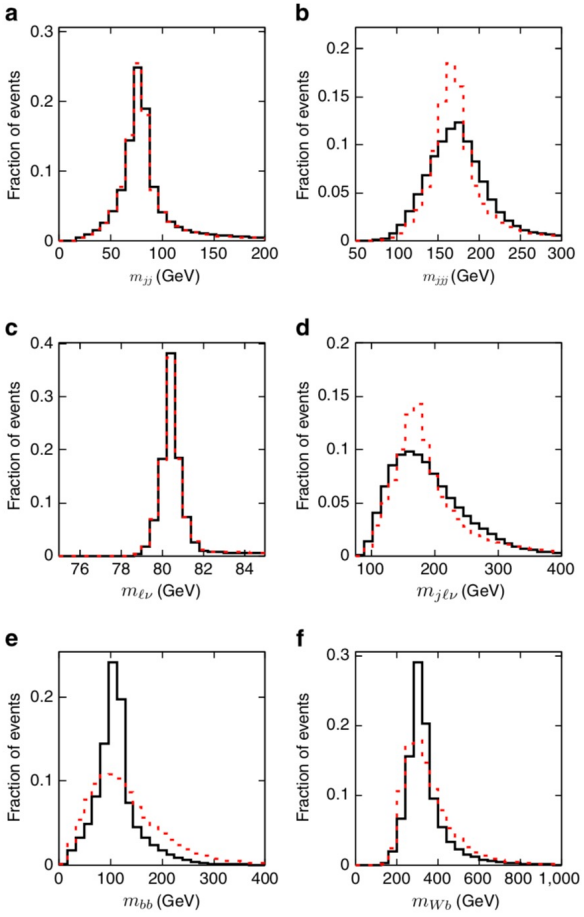
top kvarki (ozadje)



low-level quantities



high-level quantities



Domača naloga: Higgs Data Set

University of Ljubljana
Faculty of Mathematics and Physics

↳NN - 1 skrita plast (hidden layer) nevronov, Deep NN (DN, DNN) - 6 skritih plasti nevronov

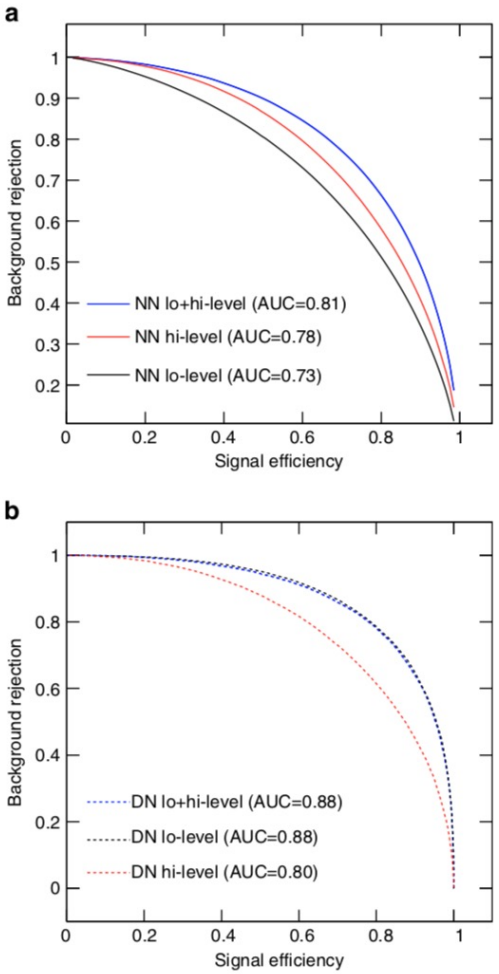


Figure 7 | Performance for Higgs benchmark. For the Higgs benchmark, comparison of background rejection versus signal efficiency for the traditional learning method (**a**) and the deep learning method (**b**) using the low (lo)-level features, the high (hi)-level features and the complete set of features.

Metrika: bkg. rejection eff. (kontaminacija) vs sig. eff. (izkoristek)

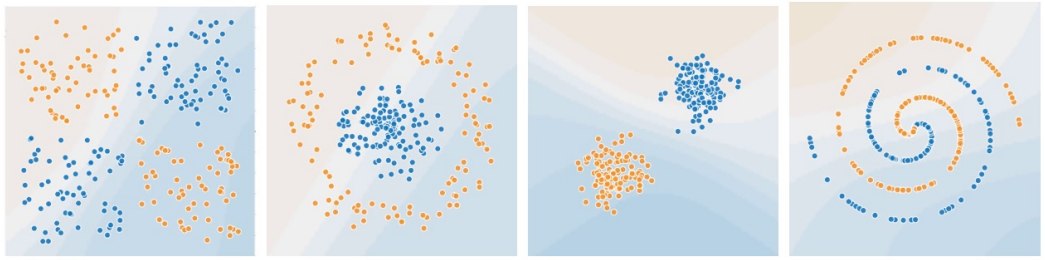
Table 1 | Performance for Higgs benchmark.

Technique	Low-level	High-level	Complete
AUC			
BDT	0.73 (0.01)	0.78 (0.01)	0.81 (0.01)
NN	0.733 (0.007)	0.777 (0.001)	0.816 (0.004)
DN	0.880 (0.001)	0.800 (<0.001)	0.885 (0.002)
Discovery significance			
NN	2.5σ	3.1σ	3.7σ
DN	4.9σ	3.6σ	5.0σ

Comparison of the performance of several learning techniques: boosted decision trees (BDT), shallow neural networks (NN), and deep neural networks (DN) for three sets of input features: low-level features, high-level features and the complete set of features. Each neural network was trained five times with different random initializations. The table displays the mean area under the curve (AUC) of the signal-rejection curve in Fig. 7, with s.d. in parentheses as well as the expected significance of a discovery (in units of Gaussian σ) for 100 signal events and 1,000 ± 50 background events.

Domača naloga: Higgs Data Set

- Implementiraj iskanje Higgsovega bozona v BDT in DNN **s pomočjo materiala na učilnici!**
  - Študiraj vpliv uporabljenih spremenljivk X!
  - Študiraj dimenzijo in nastavitve BDT (Catboost) in NN (Tensorflow):
  - Narisi training/overtraining in validacijo ter rezultat (ROC)
- Dodatna naloga: Uporabi dani metodi še na tipičnih/demo zgledih:



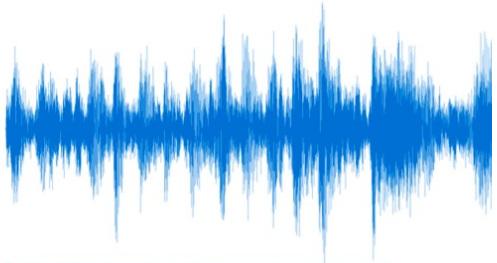
- BDT/AdaBoost v Scikit/Sklearn (če kdo hoče):
  -
```
from sklearn.ensemble import AdaBoostClassifier
ada_clf = AdaBoostClassifier(
DecisionTreeClassifier(max_depth=1), n_estimators=200,
algorithm="SAMME.R", learning_rate=0.5 )
ada_clf.fit(X_train, y_train)
```

State of the Art

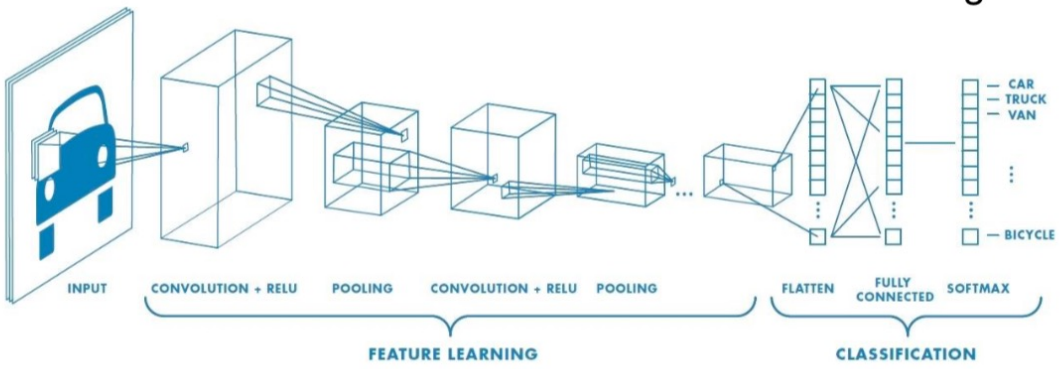
↳ Danes kombinacije različnih DNN za končni rezultat

UF UNIVERSITY of FLORIDA

Convolutional NN



Feature learning



State of the Art

↳ Generative Adversarial Networks (GAN) za simulacije...

PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION

Tero Karras
NVIDIA

Timo Aila
NVIDIA

Samuli Laine
NVIDIA

Jaakko Lehtinen
NVIDIA
Aalto University

NVIDIA

TensorFlow na IJS

University of Ljubljana
Faculty of Mathematics and Physics

- Za večje želje:
  - Google Collab: https://colab.research.google.com/
  - V okviru konzorcija SLING (http://www.sling.si/) je na voljo več gruč. Na IJS je gruča NSC, ki ima 16 kartic NVidia Kepler K40 in je uprabna za testiranje Tensorflowa. Programska oprema je že nameščena, potrebujete pa dostop preko digitalnega certifikata (overjanje) in navodila. Obrnite se na info@sling.si ali Jona.Javorsek@ijs.si

Literatura

- Literature je danes malo morje, priporočam brskanje po spletu za konkretne probleme (npr. stackoverflow.com).
- Priporočam naslednje knjige:
  - Aurélien Géron: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition
  - Ian Goodfellow and Yoshua Bengio and Aaron Courville: Deep Learning
  - Peter Flach: Machine Learning: The Art and Science of Algorithms That Make Sense of Data
  - Christopher Bishop: Pattern recognition and Machine learning
- ... pa tudi na spletni učilnici je še nekaj materiala.