

# **Uporaba metod strojnega učenja pri analizi podatkov na LHC**

Seminar - 3. letnik, I. stopnja

---

**avtor:** Jan Gavranovič

**mentor:** prof. dr. Borut Paul Kerševan

13. maj 2020

Univerza v Ljubljani, Fakulteta za matematiko in fiziko

# Kratek pregled

Uvod v LHC

Fizika na LHC-ju

Strojno učenje

Nevronske mreže

Iskanje procesa  $hh$  z uporabo nevronskeih mrež

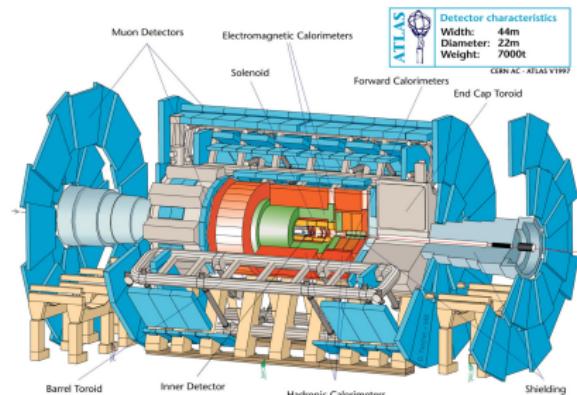
Zaključek

## **Uvod v LHC**

---

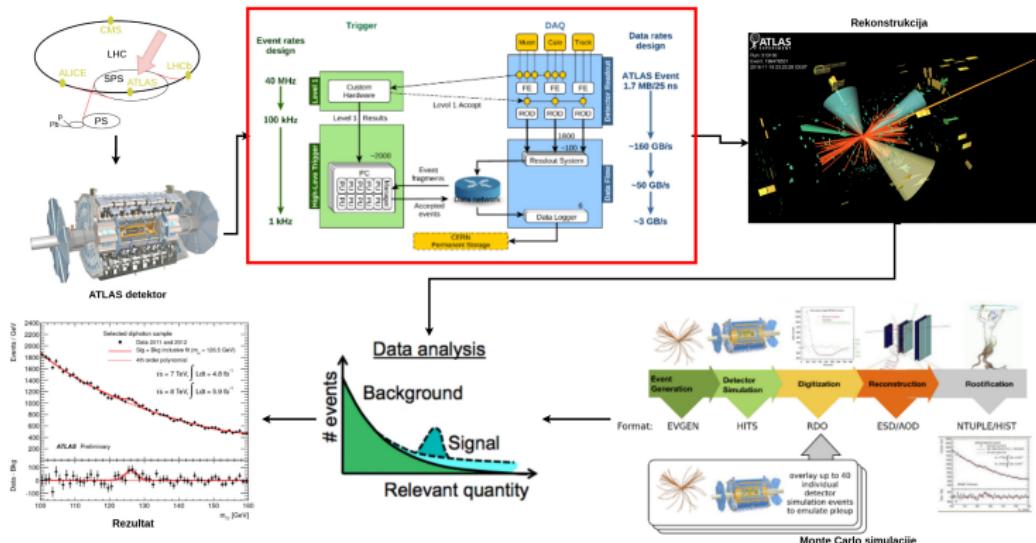
# Uvod v LHC

- LHC primarno namenjen trkanju žarkov protonov z energijami  $\sqrt{s} = 13$  TeV
- Največji pospeševalnik na svetu po velikosti in doseganju energij
- Več detektorjev - največja ATLAS in CMS
- Raziskovanje napovedi Standardnega modela in njegovim možnim razširitvam
- Izziv: analiza in interpretacija velikih količin podatkov
- Problem: Razmerje šum/signal, kompleksnost, visoka dimenzionalnost podatkov
- Rešitev: strojno učenje



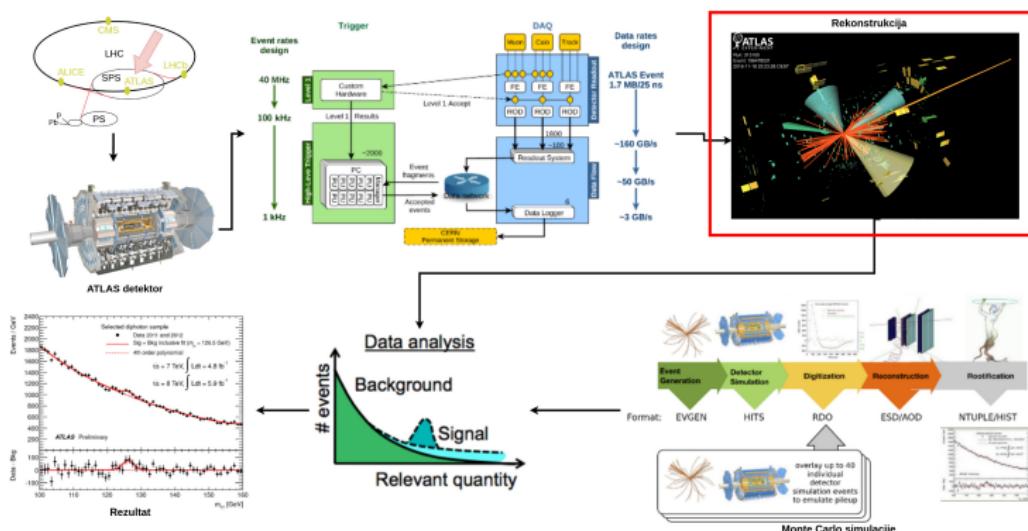
# Prožilni sistem (trigger)

- Frekvence  $\sim 40$  MHz trkanja žarkov protonov
- ATLAS ima  $\mathcal{O}(10^6)$  senzorjev  $\Rightarrow 1$  PB podatkov/s
- Prožilni sistem pove katere podatke vredno shraniti
- 2 stopnji: 1. Posebna elektronika, 2. Računalniška analiza
- Končen rezultat:  $\sim 1000$  dogodkov/s s hitrostjo prenosa podatkov  $\sim 1$  GB/s



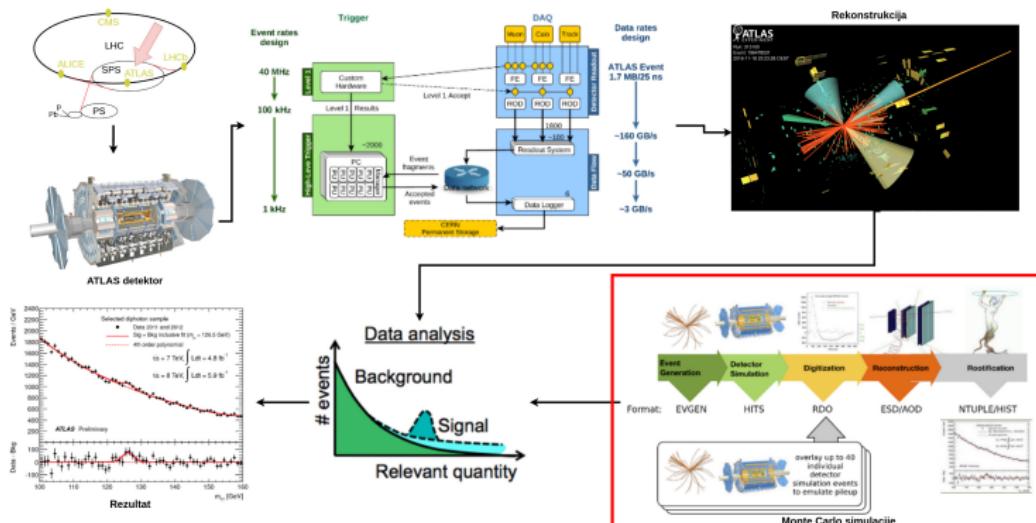
# Rekonstrukcija

- Majhne časovne skale (primer: Higgsov bozon razpade v  $\sim 10^{-22}$  s)
- Uporabni razpadni produkti začetnih delcev
- Sklepamo na delce iz katerih so nastali
- Uporaba naprednih računskih metod



# Simulacije

- Primerjava eksperimentalnih rezultatov s teoretičnimi
- Napovedovanje signalov in ozadja
- Simulacija fizikalnega procesa in odziva detektorja
- Tipa Monte Carlo

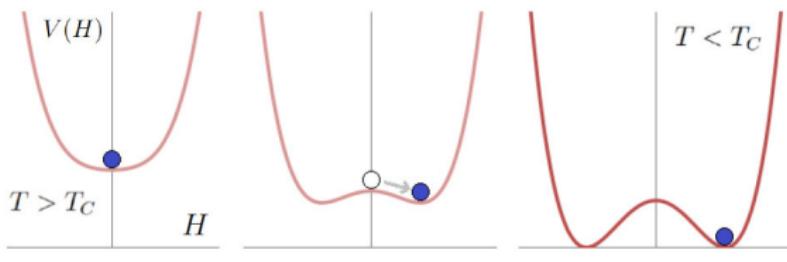


## Fizika na LHC-ju

---

# Higgsov bozon

- V Standardnem modelu Higgsov mehanizem pojasni kako bozoni dobijo maso preko zloma elektrošibke simetrije
- Brez tega mehanizma bi bili vsi delci brez mase
- Potencial Higgsovega bozona po zlomu simetrije povzroči mase delcev
- Teorija napove obstoj masivnega delca, ki je nosilec (fluktuacija) Higgsovega polja  $\Rightarrow$  Higgsov bozon
- Masa delca je odvisna od moči interakcije med tem delcem in Higgsovim poljem
- Higgsov bozon najden leta 2012 z detektorjema ATLAS in CMS, izmerjena masa je  $m_h = 125.18 \pm 0.16 \text{ GeV}/c^2$



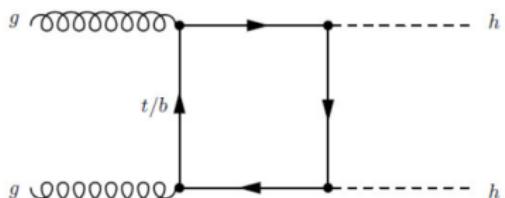
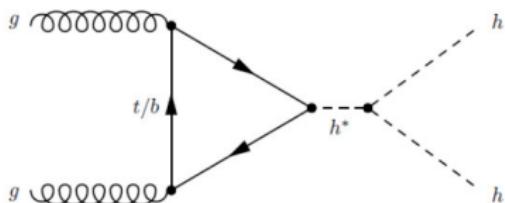
Grafičen prikaz zloma simetrije.

# Nastanek para Higgsovih bozonov

- Potencial Higgsovega bozona je:

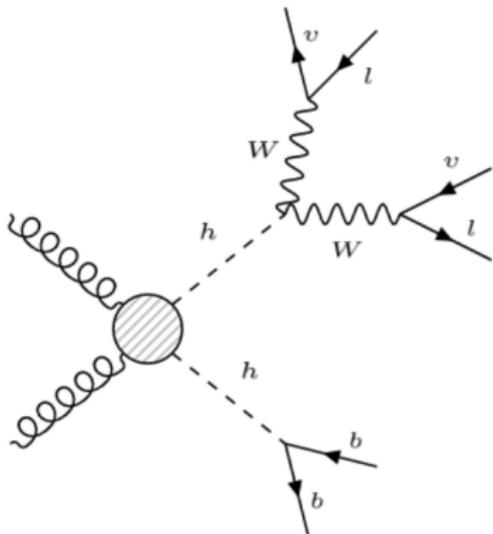
$$V(h) = \frac{1}{2} m_h^2 h^2 + \lambda v h^3 + \frac{1}{4} \lambda h^4$$

- Trilinearna sklopitevna konstanta:  $\lambda$
- Drugi člen opisuje vozlišča s tremi Higgsovimi bozoni, tretji pa s štirimi
- Naloga detektorjev LHC-ja: meritev  $\lambda$  preko procesov, ki so občutljivi na sklapljanje s tem parametrom
- Glavni način nastajanja parov poteka preko fuzije dveh gluonov
- Možnih več različnih procesov fuzije gluonov, ki ustvarijo pare  $hh$



# Nastanek para Higgsovih bozonov

- Sledi opis pred kratkim objavljene analize iskanja parov Higgsovih bozonov
- Analiza se osredotoči na razpad para  $hh \rightarrow b\bar{b}$  in  $W^+W^-$ , ki nadaljnjo razpade leptonsko v  $l\nu l\nu$
- Pred tem: opis strojnega učenja in nevronskih mrež, ki jih analiza uporablja za razločevanje med ozadjem in procesom  $hh$



## **Strojno učenje**

---

- Veja umetne inteligence (AI)
- Zasnova in razvoj algoritmov, ki lahko izboljšajo svoje vedenje na podlagi podatkov
- Cilj: samodejno učenje prepoznavanja vzorcev in sprejemanje intelligentnih odločitev

## Definicija algoritemskega učenja

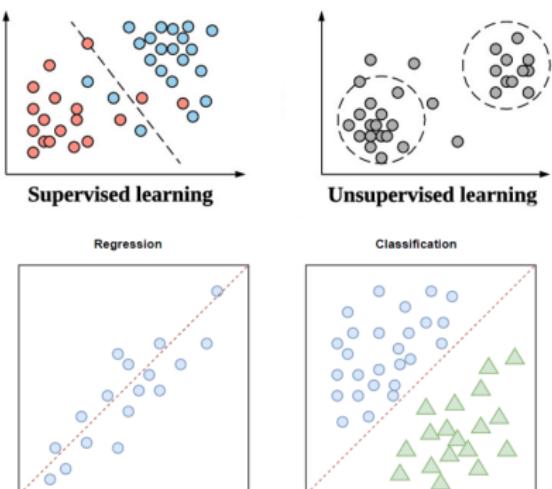
Računalniški program se uči iz izkušenj E glede na nalogu T in na neko mero uspešnosti P, če se uspešnost za T, ki jo meri P poveča z izkušnjami E.

Dve širši skupini algoritmov:

### 1. Nadzorovano učenje

- klasifikacija (diskretna kategorizacija podatkov)
- regresija (modeliranje, prilagajanje)

### 2. Nenadzorovano učenje



Problem nadzorovanega učenja je sestavljen iz:

- množice podatkov  $\mathcal{D}_{data} = \{\mathbf{x}_i, y_i\}_{i=1}^N$  sestavljene iz meritev  $\mathbf{x}_i$  in ciljnih vrednosti  $y_i$  vzorčenih iz verjetnostne porazdelitve  $p_{data}(y|\mathbf{x}; \theta)$ , kjer je  $\{\theta_i\}$  množica parametrov modela
- optimizacijskega algoritma
- učnega algoritma, ki vrne funkcijo  $h : \mathbf{x} \rightarrow y$ , funkcija  $h(\mathbf{x}; \theta)$  je napoved modela
- funkcije izgube  $L(y_i, h(\mathbf{x}_i; \theta))$  ter funkcije cene:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N L(y_i, h(\mathbf{x}_i; \theta))$$

Cilj metod strojnega učenja je iskanje različnih parametrov, uteži ter struktur, ki minimizirajo tako funkcijo.

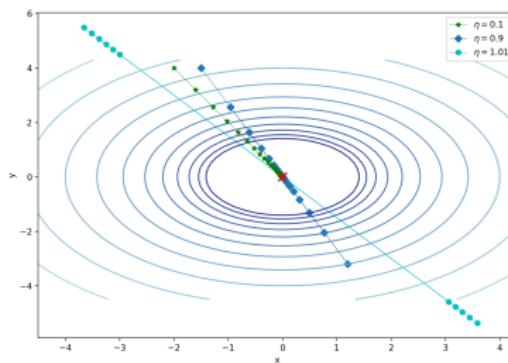
# Optimizacijski algoritmi

- Poiskati želimo vrednost  $\theta$ , ki minimizira funkcijo cene  $J(\theta)$
- Razredno metod, ki minimizira  $J(\theta)$  pravimo *gradient descent*
- Ideja enaka kot pri Newtonovi metodi
- Pri osnovni verziji GD algoritma začnemo z  $\theta_0$  in iterativno posodabljammo:

$$\mathbf{v}_i = \eta \nabla_{\theta} J(\theta_i)$$

$$\theta_{i+1} = \theta_i - \mathbf{v}_i$$

- Konvergenca odvisna od izbire velikosti koraka  $\eta$  (*learning rate*)



- GD se ponavadi ne uporablja zaradi počasne konvergencije in drugih težav
- Uporabljajo pa se njegove nadgradnje, kot je Adam

## Mini-batch gradient descent

- Potreben izračun:

$$\nabla_{\theta} J(\theta) = \frac{1}{N} \sum_{i=1}^N \nabla_{\theta} L(y_i, \mathbf{x}_i; \theta)$$

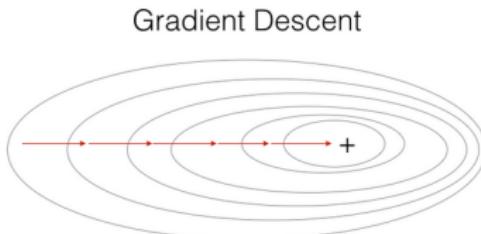
- Cena operacije  $\mathcal{O}(N)$
- Podatke razdelimo v skupine (*mini-batches*) velikosti  $M$ :

$$B = \{(\mathbf{X}^{\{1\}}, y^{\{1\}}), \dots, (\mathbf{X}^{\{K\}}, y^{\{K\}})\}$$

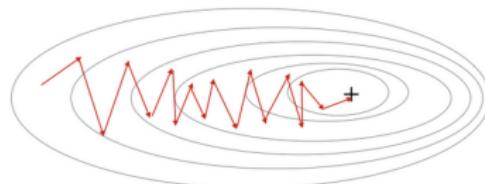
- Približek za gradient je tako

$$\nabla_{\theta} J(\theta) \approx \frac{1}{M} \sum_{i=1}^M \nabla_{\theta} L(y_i, \mathbf{x}_i; \theta)$$

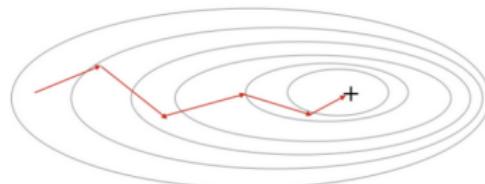
- V vsakem koraku porabimo eno skupino  $B_k$
- Iteraciji čez vse podatke N, torej čez vse skupine  $B_k$ , pravimo epoha



Gradient Descent



Stochastic Gradient Descent



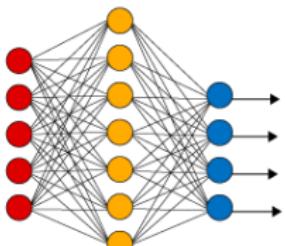
Mini-Batch Gradient Descent

## Nevronske mreže

---

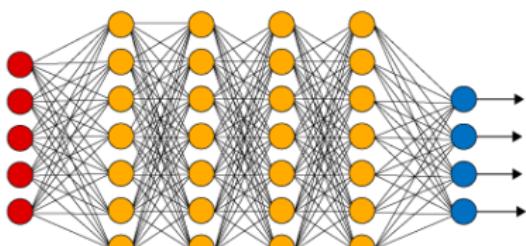
- Vrsta nadzorovanega strojnega učenja
- Nedavni uspeh strojnega učenja je prišel predvsem iz uporabe velikih količin podatkov za učenje globokih nevronskeih mrež, ki so sestavljene iz velike količine nelinearnih plasti (*deep learning*)
- Globlje mreže lahko predstavijo kompleksne funkcije bolj učinkovito
- Napredki v strojni opremi in izboljšave algoritmov v zadnjih letih
- V eksperimentalni fiziki ne manjka podatkov, ki bi bili uporabni za učenje nevronskeih mrež

**Simple Neural Network**



● Input Layer

**Deep Learning Neural Network**



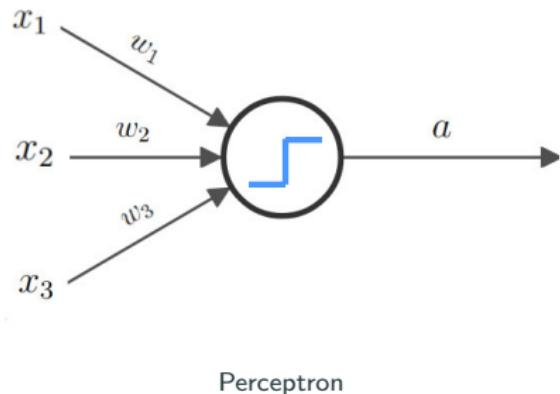
● Hidden Layer

● Output Layer

- Osnovna enota nevronske mreže
- Sprejme vektor lastnosti  $x$  in vrne skalar  $a(x)$
- Funkcija  $a$  sestavljena iz linearne operacije in nelinearne transformacije

$$a = \sigma(z) = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

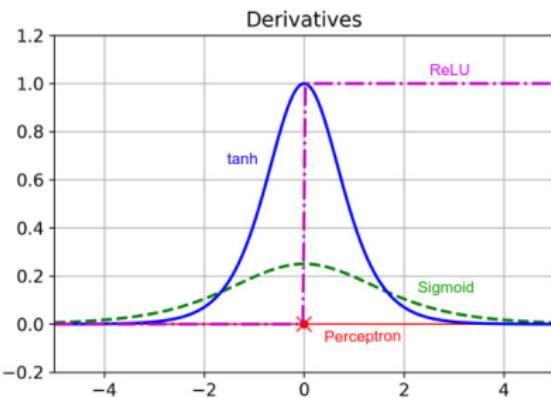
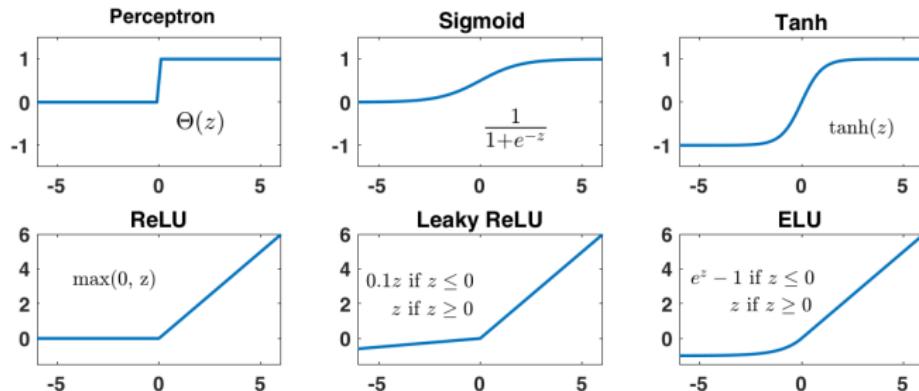
- Različne izbire aktivacijskih funkcij vodijo do različnih računskih sposobnosti nevronov
- Danes se večinoma uporablja aktivacijske funkcije tipa ReLU



$$a = \begin{cases} 0 & \text{če } \mathbf{w} \cdot \mathbf{x} \leq \text{aktivacijski prag} \\ 1 & \text{če } \mathbf{w} \cdot \mathbf{x} > \text{aktivacijski prag} \end{cases}$$

- Definiramo:  $b \equiv -\text{aktivacijski prag}$
- Problem z nasičenjem

## Primeri aktivacijskih funkcij in njihovi odvodi



# Usmerjena in povratna propagacija

- Usmerjena propagacija: povezava med plasti  $l$  in  $l - 1$

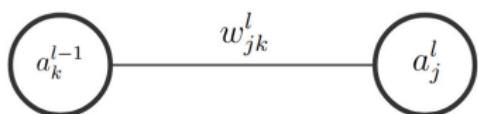
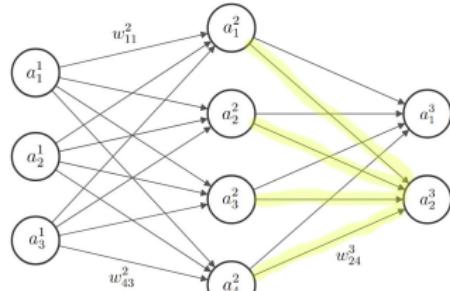
$$a_j^l = \sigma \left( \sum_k w_{jk}^l a_k^{l-1} + b_j^l \right) = \sigma(z_j^l)$$

- Povratna propagacija: učinkovit izračun odvodov

$$\frac{\partial J}{\partial w_{jk}^l} \quad \text{in} \quad \frac{\partial J}{\partial b_j^l}$$

- Več plasti, zaradi katerih je računanje gradienta bolj zapleteno
- Za funkcijo izgube lahko uporabimo MSE

$$J(a^L) = \frac{1}{2} \sum_j (y_j - a_j^L)^2$$



- Vpeljemo vmesno količino, ki predstavlja napako  $j$ -tega nevrona v plasti  $I$

$$\Delta_j^I = \frac{\partial J}{\partial z_j^I}$$

- Povratno propagacijo sestavljajo 4 osnovne enačbe preko katerih izračunamo gradient funkcije izgube
- Izpeljemo s pomočjo verižnega pravila po plasteh

$$\Delta_j^L = \frac{\partial J}{\partial a_j^L} \sigma'(z_j^L) \quad (1)$$

$$\Delta_j^I = \sum_k w_{kj}^{I+1} \Delta_k^{I+1} \sigma'(z_j^I) \quad (2)$$

$$\frac{\partial J}{\partial b_j^I} = \Delta_j^I \quad (3)$$

$$\frac{\partial J}{\partial w_{jk}^I} = \Delta_j^I a_k^{I-1} \quad (4)$$

# Učenje nevronske mreže

**Data:** Podatki  $\mathbf{X}$  in oznake  $\mathbf{y}$  razdeljeni v skupine.

**Result:** Optimalne uteži  $W^l$  in biase  $b^l$  nevronske mreže.

Naključno inicializiraj vse uteži  $W^l$  in biase  $b^l$  po vseh plasteh;

**for epoch do**

**for mini-batch  $B_k$  do**

**for primer  $(\mathbf{x}_i, \mathbf{y}_i)$  iz  $B_k$  do**

            1. Nastavi prvo aktivacijo  $a^{i,1}$ .

            2. Izvedi usmerjeno propagacijo po plasteh  $l = 2, 3, \dots, L$ .

            3. Izračunaj napako v zadnji plasti  $\Delta^{i,L}$  po (1).

            4. Propagiraj napake  $\Delta^{i,l}$  po plasteh  $l = L, L-1, \dots, 2$  z (2).

**end**

    Uporabi optimizacijski algoritem npr. GD ter za vse  $l = L, L-1, \dots, 2$

    posodobi uteži in biase z uporabo (3) in (4) kot

$$W^l \rightarrow W^l - \frac{\eta}{M} \sum_{i=1}^M \Delta^{i,l} (\mathbf{a}^{i,l-1})^T \quad \text{in} \quad b^l \rightarrow b^l - \frac{\eta}{M} \sum_{i=1}^M \Delta^{i,l} .$$

**end**

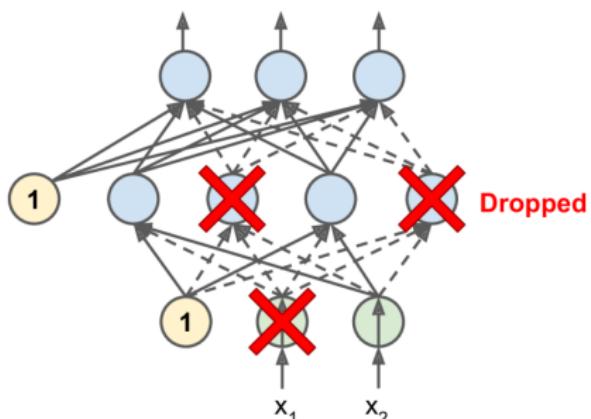
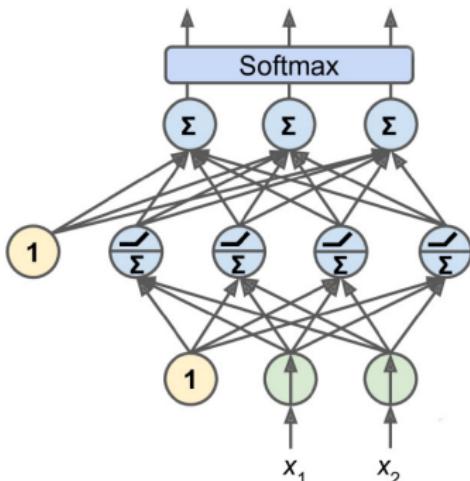
**end**

# Plast softmax in dropout regularizacija

- V zadnji plasti lahko uporabimo plast softmax

$$a_j^L = \frac{e^{z_j^L}}{\sum_k e^{z_k^L}}$$

- Velja:  $\sum_j a_j^L = 1$
- Izhodi nevronske mreže so na intervalu  $[0, 1]$  in se seštejejo v 1
- Predstavlja verjetnostno porazdelitev  $\Rightarrow a_j^L$  verjetnost, da je pravilen razred  $j$
- Dropout plast: skrita plast v kateri naključno izbrišemo neko število nevronov
- Postopek zmanjša *overfit* (manjše število prostih parametrov)

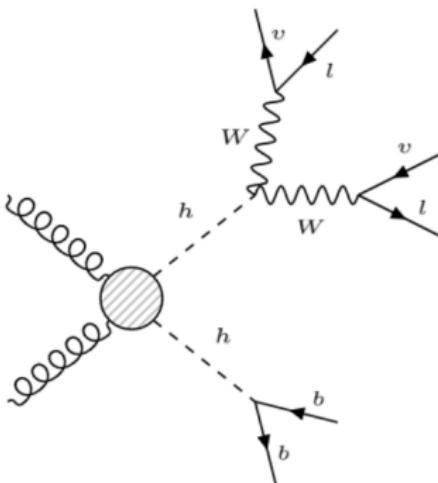


## **Iskanje procesa $hh$ z uporabo nevronskeih mrež**

---

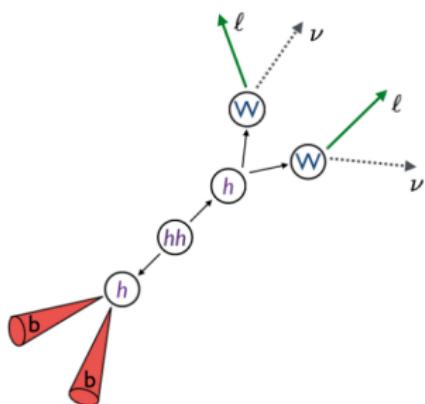
## Iskanje procesa $hh$ z uporabo nevronskeih mrež

- Analiza obravnava razpad para Higgsovih bozonov v  $WWb\bar{b} \rightarrow l\nu l\nu b\bar{b}$
- Uporabljenih  $139 \text{ fb}^{-1}$  podatkov trkov  $pp$  pri  $\sqrt{s} = 13 \text{ TeV}$  zbranih na detektorju ATLAS
- Procesi ozadja in signala iz Standardnega modela ocenjeni preko Monte Carlo simulacij
- Analiza je tipa MVA (*multivariate analysis*)
- Cilj analize je bil določiti zgornjo mejo sipalnega preseka za  $pp \rightarrow hh$  proces ter postaviti meje intervala za skloplitveno konstanto  $\lambda$

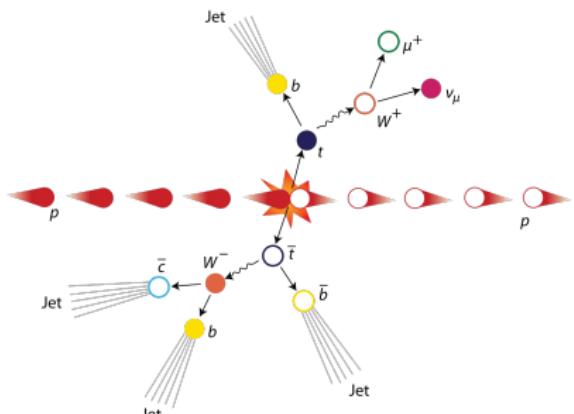


# Arhitektura in učenje nevronske mreže

- Klasifikacija v štiri razrede
  1. Tvorba  $hh$
  2. Tvorba kvarkov top:  $t\bar{t}$  in  $Wt$
  3. Drell-Yan proces z lahkimimi leptoni:  $Z/\gamma^* \rightarrow ll$
  4. Drell-Yan proces z leptoni tau:  $Z/\gamma^* \rightarrow \tau\tau$
- Za vhodne podatke uporabili 35 enostavnih kot tudi kompleksnejših količin
- Učenje nevronske mreže poteka na simuliranih dogodkih iz Standardnega modela

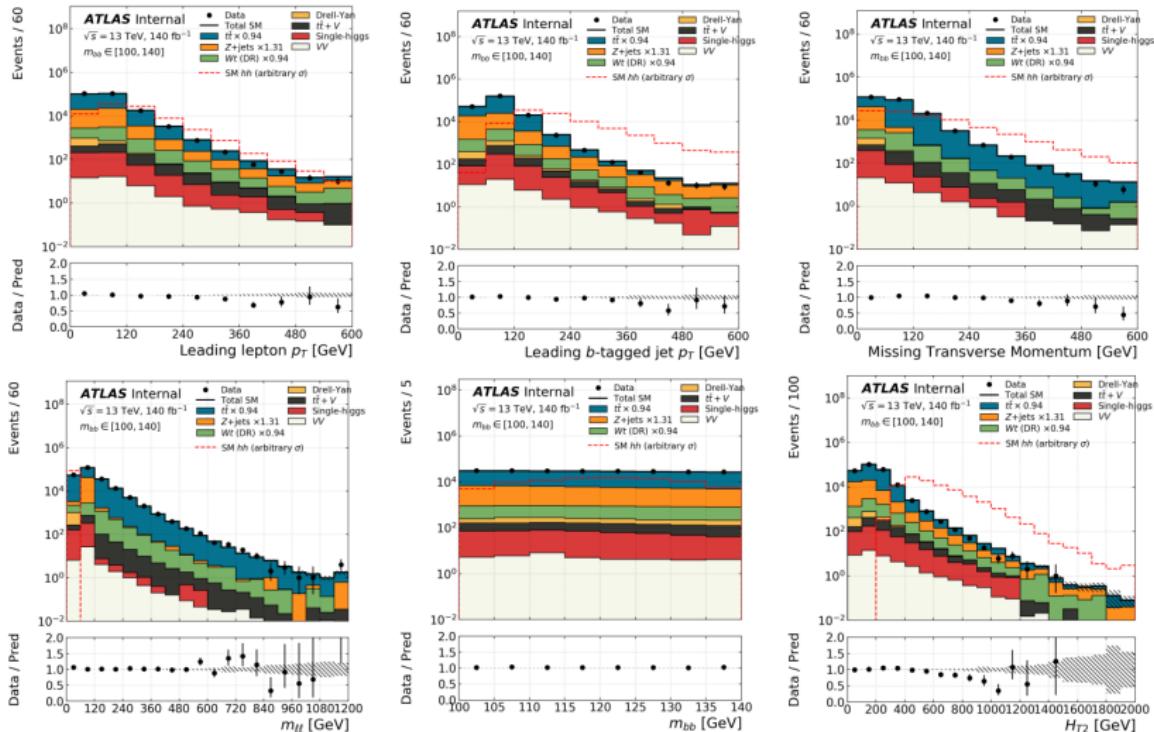


Signal



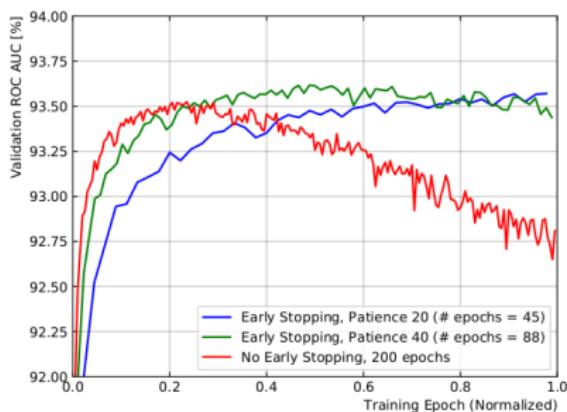
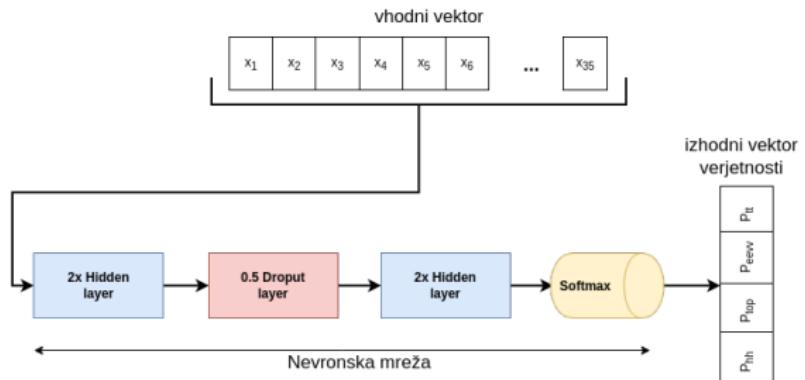
Ozadje

# Arhitektura in učenje nevronske mreže



Porazdelitve nekaterih vhodnih kinematičnih spremenljivk nevronske mreže. V prvi vrstici so enostavne količine. V drugi vrstici pa so sestavljeni količini.

# Arhitektura in učenje nevronske mreže

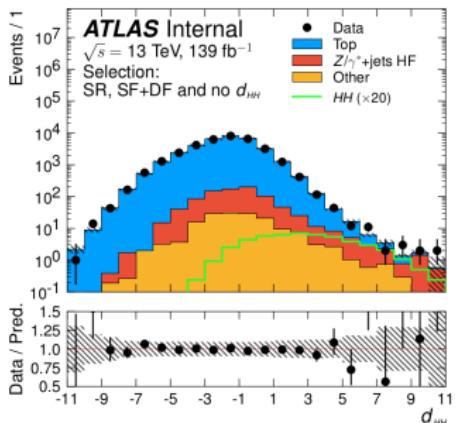
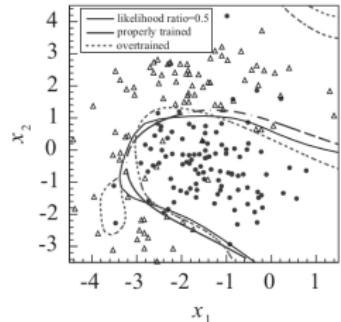
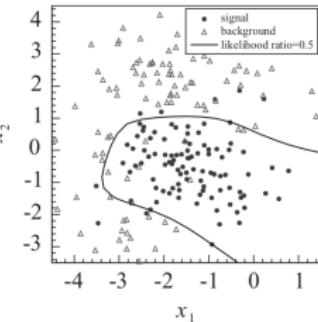


# Izbira območja signala

- Nevronska mreža vrne verjetnosti za štiri procese:  $[p_{hh}, p_{top}, p_{Zsf}, p_{Z\tau\tau}]$
- Iz tega rezultata analiza sestavi diskriminanto

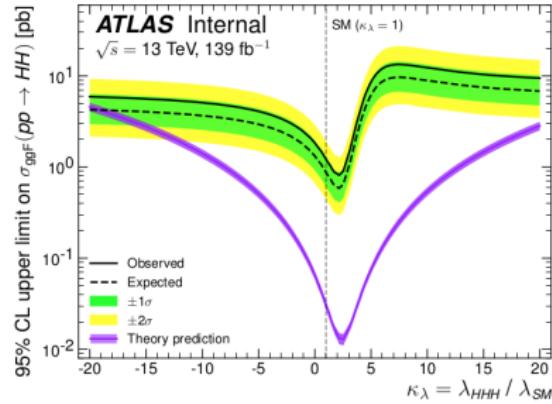
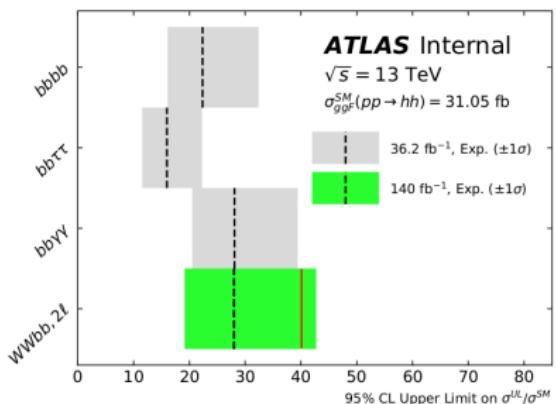
$$d_{hh} = \ln \left( \frac{p_{hh}}{p_{top} + p_{Zsf} + p_{Z\tau\tau}} \right)$$

- Vsebuje informacijo o zmožnosti nevronske mreže, da loči med štirimi procesi
- Območje signala je definirano glede na primerno vrednost  $d_{hh}$  (*cut and count* pristop)



# Rezultati analize

- Kinematično prilagajanje napovedi za signal in ozadje na izmerjene podatke
- Določili s stopnjo zaupanja (*confidence level, CL*) 95% zgornja mejo (*upper limit, UL*) sipalnega preseka tvorbe para  $hh$
- Izmerjena zgornja meja preseka za faktor  $29^{+14}_{-9}$  višja od napovedane po Standardnem modelu
- Izračunali sipalni presek pri različnih vrednostih  $\kappa_\lambda = \lambda_{HHH} / \lambda_{SM}^{SM}$
- Možnosti teorij izven Standardnega modela  $\Rightarrow \kappa_\lambda \neq 1$
- Dovoljena vrednost  $\kappa_\lambda$  pri stopnji zaupanja 95% leži na intervalu  $[-19, > 20]$

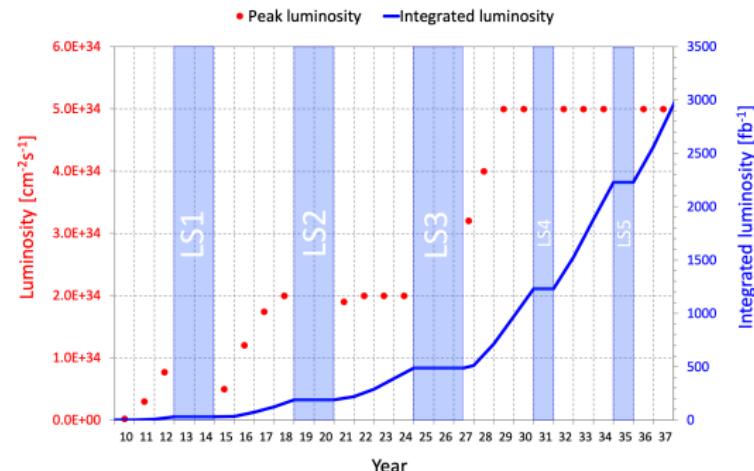


## Zaključek

---

# Zaključek

- Nadaljnji razvoj algoritmov in računalniške opreme ter vse več podatkov
- LHC se nadgrajuje še naprej → HL-LHC
- Veliko število dogodkov ter s tem povezane velike količine podatkov, ki jih bo potrebno obdelati in analizirati



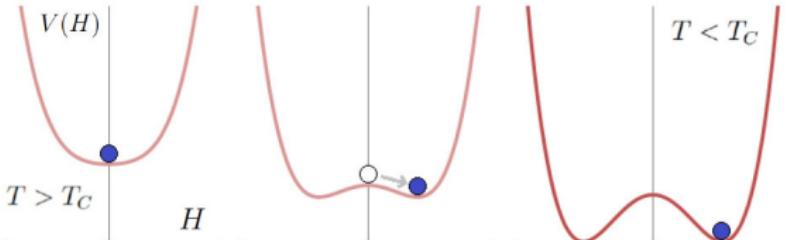
(HL-)LHC napoved glede na nominalne parametre.

## **Dodatek**

---

## Spontan zlom elektrošibke simetrije

- V zgodnjem vesolju,  $T > T_C$ , je imelo Higgsovo polje povprečno vrednost 0 povsod
- Elektromagnetna in šibka sila združeni v elektrošibko silo
- Pri nadaljnjem širjenju vesolja se elektrošibka sila razcepi na dva dela: elektrošibka simetrija je zlomljena
- Higgsovo polje dobi pri  $T < T_C$  neničelno povprečno vrednost povsod
- Vrednosti se reče vakuumska pričakovana vrednost,  $v$
- Se zgodi zaradi točno določenega potenciala
- Masa delca je odvisna od moči interakcije med tem delcem in Higgsovim poljem

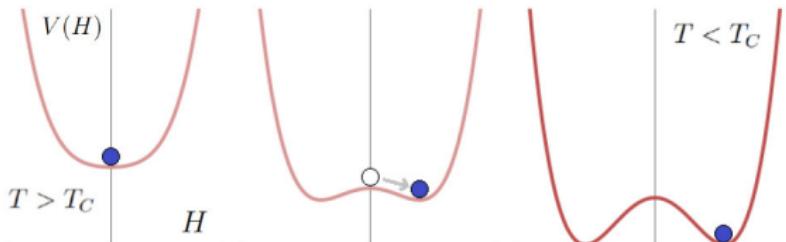


## Masa in skapljanje Higgsovega bozona

- Maso Higgsovega bozona, ki je fluktuacija Higgsovega polja  $H$  okoli ravovesne vrednosti definira Higgsov potencial  $V(H)$
- Polje  $H$  razstavimo na konstanten del  $v$  ter na del  $h$ , ki niha okrog ravovesne vrednosti in predstavlja Higgsov bozon
- Potencial Higgsovega bozona je tako

$$V(h) = \frac{1}{2} m_h^2 h^2 + \lambda v h^3 + \frac{1}{4} \lambda h^4$$

- Zadnje dva člena ustreza skapljanju Higgsovega bozona samega s seboj
- $\lambda$  ... trilinearna sklopitvena konstanta
- $\lambda$  določa obliko potenciala in s tem naravo zloma elektrošibke interakcije



## Funkcija cene

- V strojnem učenju se definira funkcija cene kot

$$J(\theta) = \mathbb{E}_{(x,y) \sim \hat{p}_{\text{data}}} L(y, h(x; \theta)) = \frac{1}{N} \sum_{i=1}^N L(y_i, h(x_i; \theta))$$

- Pogosto se za funkcijo izgube uporablja križna entropija
- Za poljubni verjetnostni porazdelitvi  $p(x)$  in  $q(x)$  jo napišemo kot

$$H(p, q) = - \sum_i p(x_i) \log q(x_i)$$

- V teoriji strojnega učenja je definirana med empirično  $\hat{p}_{\text{data}}$  in modelsko  $p_{\text{model}}$  porazdelitvijo
- Se uporablja kot funkcija izgube pri nevronskih mrežah za klasifikacijo

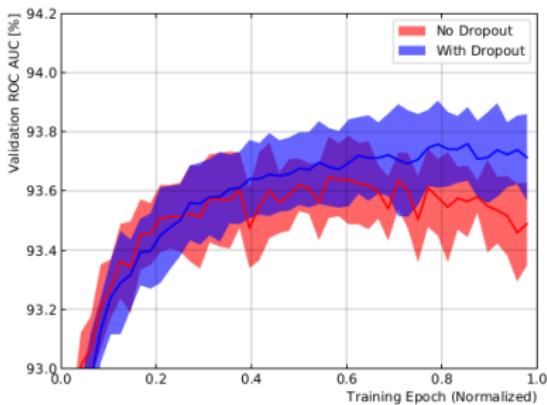
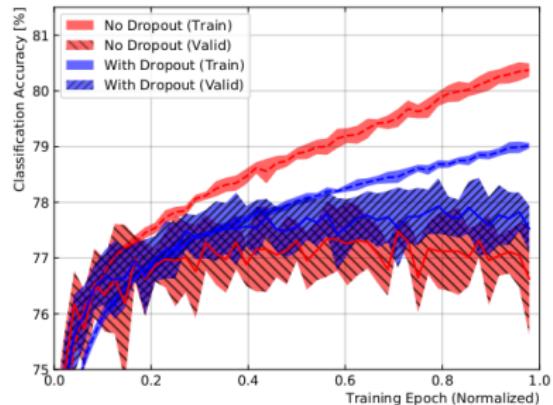
$$J(a^L) = - \sum_j y_j \log a_j^L + (1 - y_j) \log(1 - a_j^L)$$

- *Cross-entropy* med empirično porazdelitvijo  $\hat{p}_{\text{data}} = \{\hat{y}, \hat{y} - 1\}$  in modelsko porazdelitvijo  $p_{\text{model}} = \{y, y - 1\}$

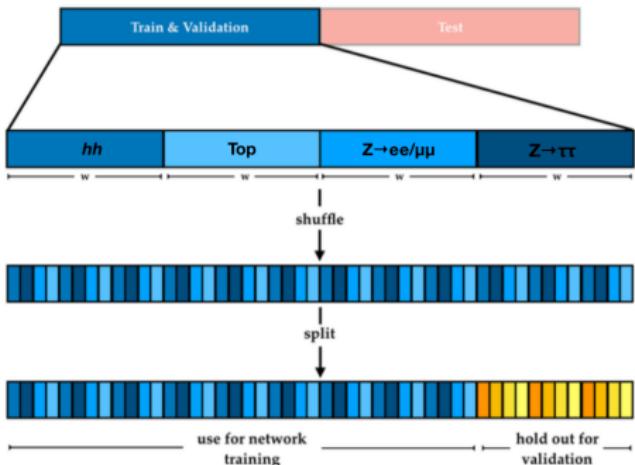
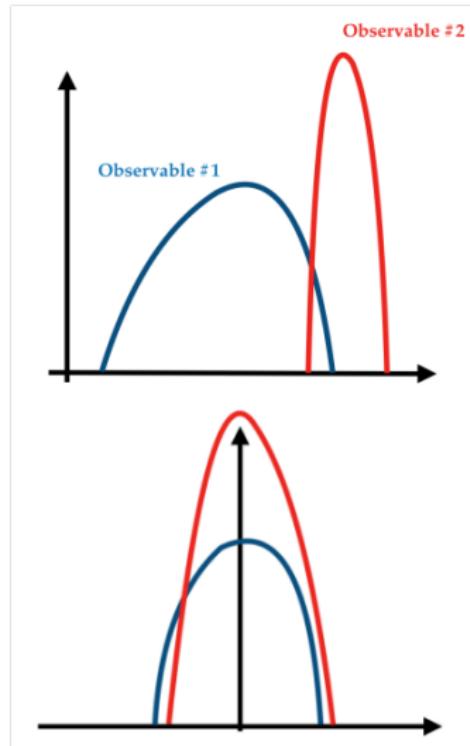
5 0 4 1 9 2 1 3 1 4  
3 5 3 6 1 7 2 8 6 9  
4 0 9 1 1 2 4 3 2 7  
3 8 6 9 0 5 6 0 7 6  
1 8 7 9 3 9 8 5 9 3  
3 0 7 4 9 8 0 9 4 1  
4 4 6 0 4 5 6 1 0 0  
1 7 1 6 3 0 2 1 1 7  
9 0 2 6 7 8 3 9 0 4  
6 7 4 6 8 0 7 8 3 1

Za ta primer bi imela prva plast  $28 \times 28$  nevronov (število pixlov) ter izhodna plast 10 nevronov (številke 0 do 9). Pravilno številko (ciljno vrednost) opiše vektor dolžine 10 samih ničel in ene enice (*one-hot vector*) na mestu, ki označuje pravilen razred.

# Dropout plast uporabljena v analizi



## Cross-validation in preprocessing



Premik spremenljivke  $x \rightarrow \frac{x-\mu}{\sigma^2}$  ter delitev vhodnih podatkov pri analizi.

## Curse of dimensionality



Ena dimenzija: N primerov, d dimezij:  $\mathcal{O}(N^d)$  primerov