

ORACLE

Machine Learning – Deep Learning – Oracle perspektívák

A Gépi Tanulás a vállalati informatikában

Molnár Balázs

Cloud Architect

Október, 2019

Safe harbor statement

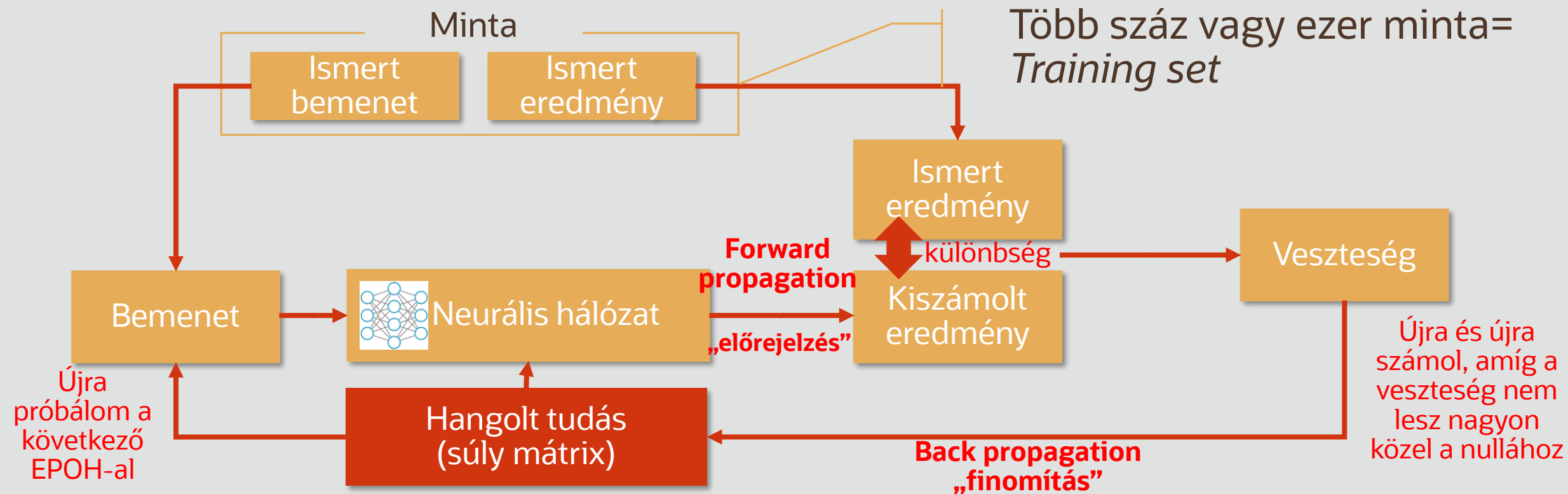
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Az új Mesterséges Intelligencia és gépi tanulás paradigmája



Hogyan tanul?



A hálózatokat ezernyi ismert bemenet/kimenet párral, mintával tanítjuk

A hálózat a „tudását” fejleszti a kiszámolt és az ismert kimenet közti különbség alapján

De lássuk inkább ezt élőben : [Tensorflow Playground](#)

Hogyan tanul?

Fontos, hogy legyen elég mintánk megfelelő formátumban

Minta

Ismert bemenet

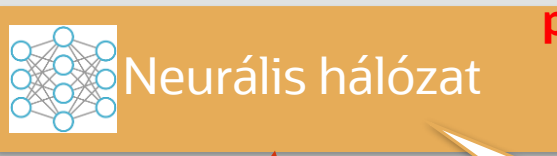
Ismert eredmény

Több száz vagy ezer minta = *Training set*

A rendszer tudását a hálózat architektúrája és a tanított súlymátrix adja

Ismerni fogjuk a rendszer várható pontosságát

Bemenet



Forward propagation

„előrejelzés”

Ismert eredmény

Kiszámolt eredmény

különbség

Veszteség

Újra és újra számol, amíg a veszteség nem lesz nagyon alacsony

Újra próbálom a következő EPOCH-al

Hangolt tudás (súly mátrix)

A neuronok száma függ a be és kimeneti adat nagyságától, illetve a szabály összetettségétől

Back propagation

Meglehetősen sokszor kell végrehajtani a forward és back propagation funkciót minden neuron számára

A hálózatokat ezernyi ismert bemenet

A hálózat a „tudását” fejleszti a kiszám

De lássuk inkább ezt élőben : [Tensorflow Playground](#)

Mitől „Deep” a Deep Learning?

Több adatfeldolgozó réteg

A neurális hálókba kerülő adatok előkészítésére, egyszerűsítésére szolgál

Réteg típusok

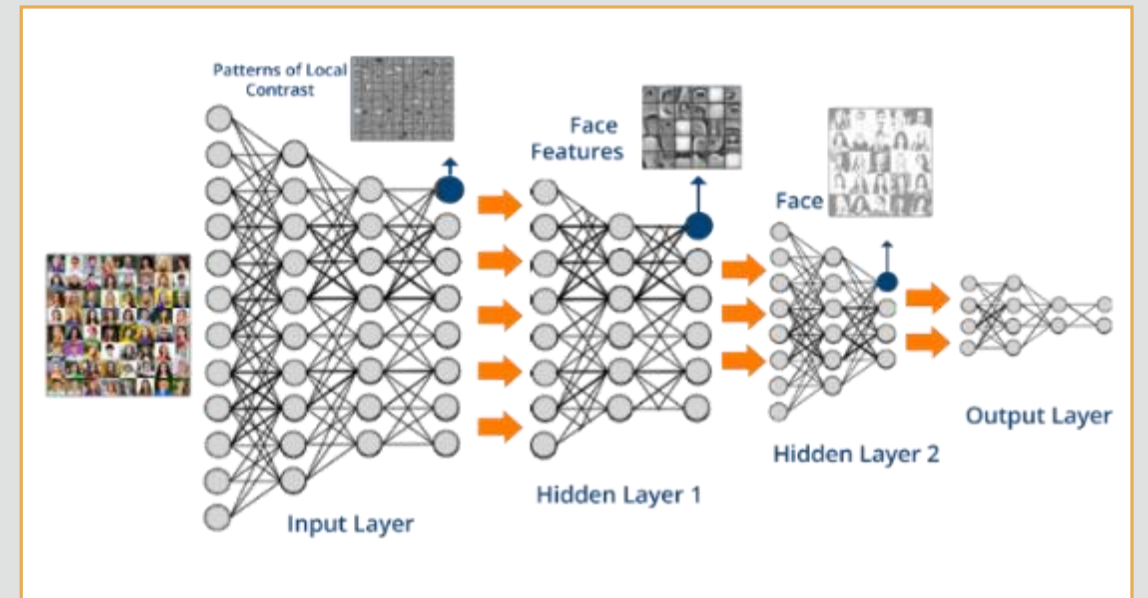
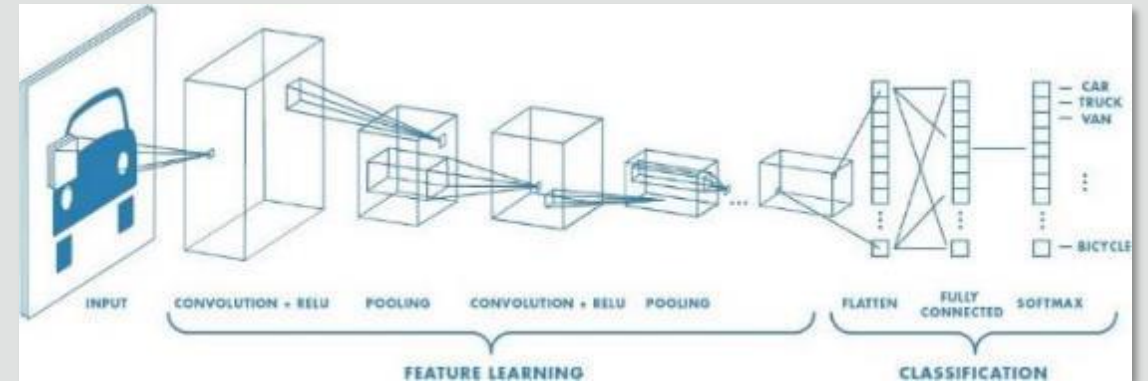
Konvolució réteg – Pl. kép filterezés

Pooling réteg – egyszerűsíti az adatot és csökkenti a szükséges neuronok számát

Teljesen összekapcsolt rétegek. (Klasszikus neurális hálók)

Stb. stb.

A „rétegrend” kialakítása is az adattudósok feladata



GAN* – M.I. generált arcok

A Neurális hálókat meg is fordíthatjuk.
Néhány alapvető attribútum alapján
(lakóhely, hajszín, nem stb.) valóság-hű
arcokat tud generálni

Az itt látható arcok egyike sem valós.
Neurális hálózatokkal generálták Őket.

*Generative Adversarial Network

Source: <https://futurism.com/incredibly-realistic-faces-generated-neural-network>



Tehát mire is képes a neurális háló?

Képes választ adni minden kérdésre...

- Bizonyos pontossággal, amennyiben van kellő számú mintánk a helyes vagy helytelen válaszokra

Osztályozni üzleti entitásokat

- Melyek a legjobb és legrosszabb beszállítók?
- Melyek a legértékesebb és legkevésbé értékes ügyfelek?
- Mely ügyfelek hagynak ott minket és miért?
- Kit vegyünk fel az üres pozícióra?
- Normális az adatbázis motorunk működése, vagy hiba várható?

Regresszió

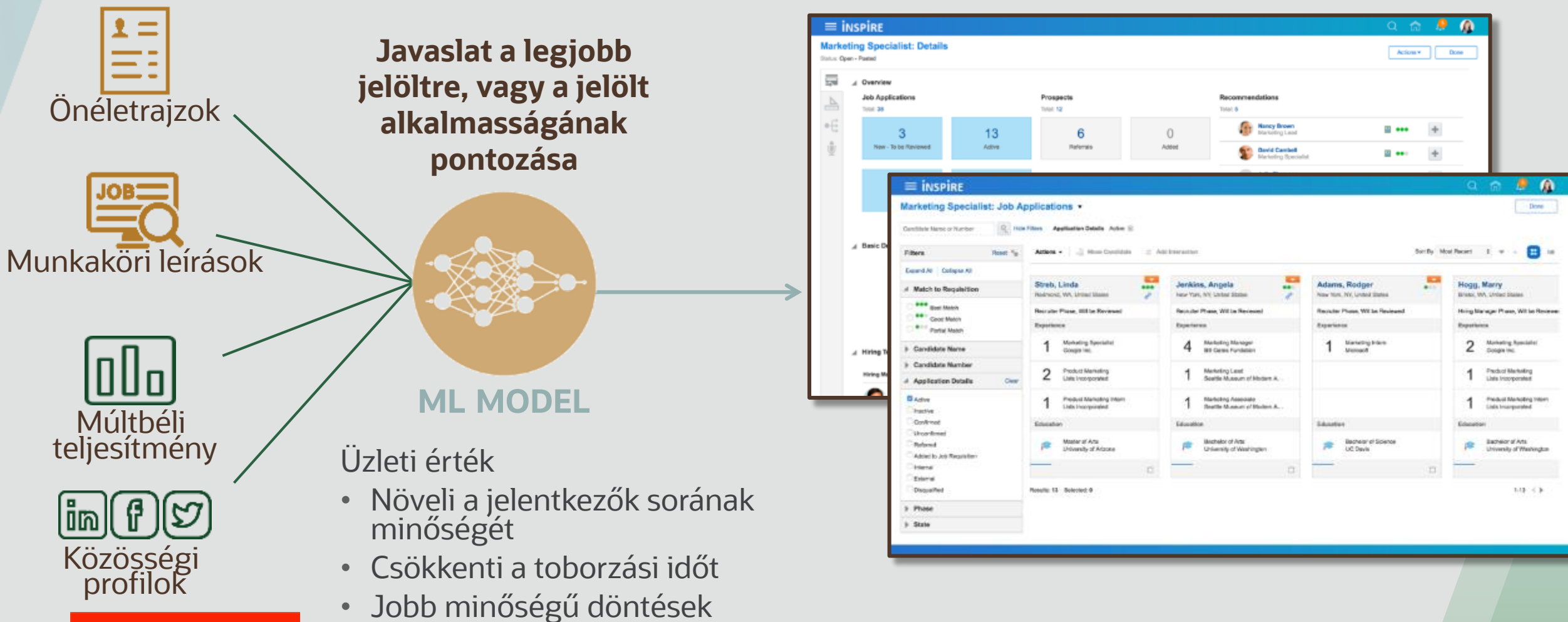
- Mekkora lesz az infláció a következő hónapban? Milyen részvényt vegyek?

Adat generálás

- Ideális ügyfélparaméterek, Ideális konfigurációs paraméterek
- Munkaköri leírás, szerződések szövege, alapadatok alapján (emberi lektorálással)
- Program Kód?

Példa: Oracle Human Capital Management

Jelentkező/állás összerendelés, jelentkezők oszályozása a feladatra



További Példák

- **Oracle ERP Cloud: Beszállítók értékelése** múltbéli teljesítményük, hírek, részvényárfolyamok, közösségi média alapján
- **Javaslat tevő motorok: Árúk, szolgáltatások** javaslata az ügyfélprofil (vásárolt termékek, aktivitás) alapján.
- **Gyártási minőség előrejelzése** IoT adatok, emberi tényezők, megrendelés összetétel, vagy alapanyag alapján
- **Hangulat index:** Egy adott szövegben (pl. cikk) egy adott témával kapcsolatos érzelmek osztályozása

DL keretrendszerek pl. Tensorflow

Nyílt forráskódú könyvtárak a gépi tanulás illetve a Deep learning támogatására

Komplex mátrixműveletek támogatása

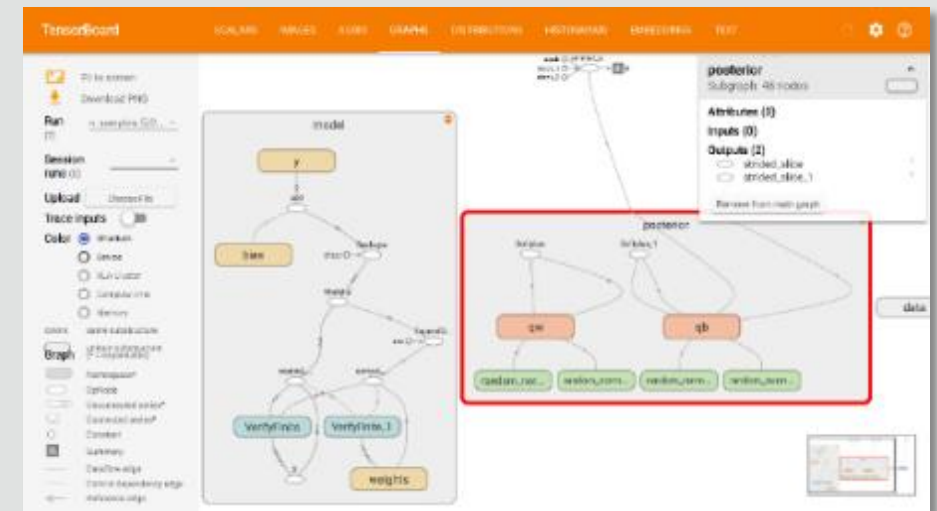
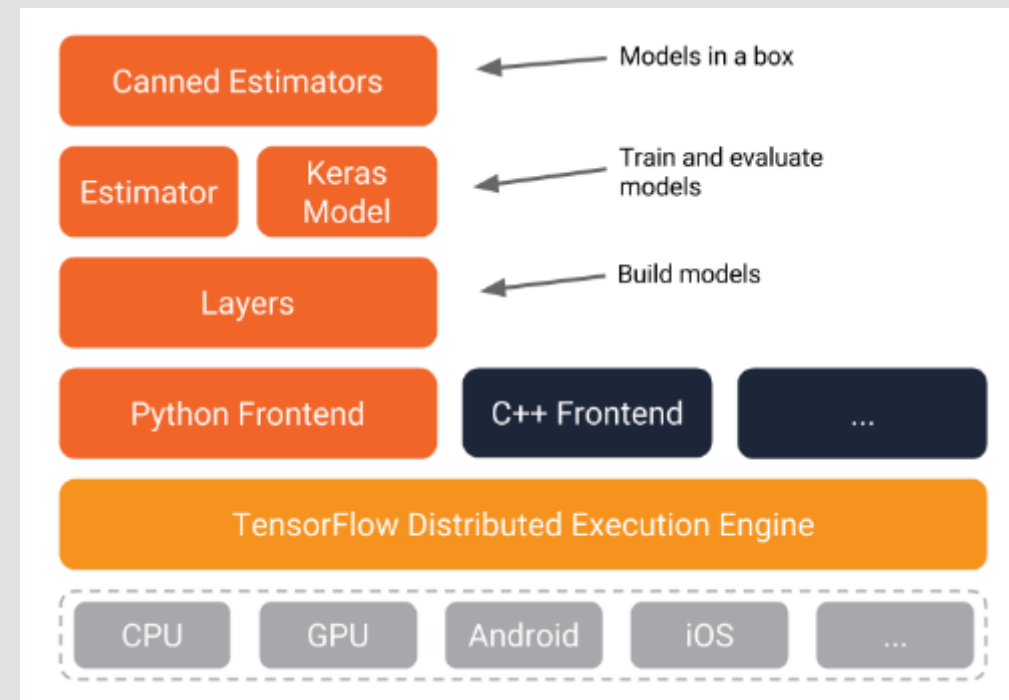
Python és C++ API

Grafikus eszközökkel, folyamatként lehet definiálni a számításokat

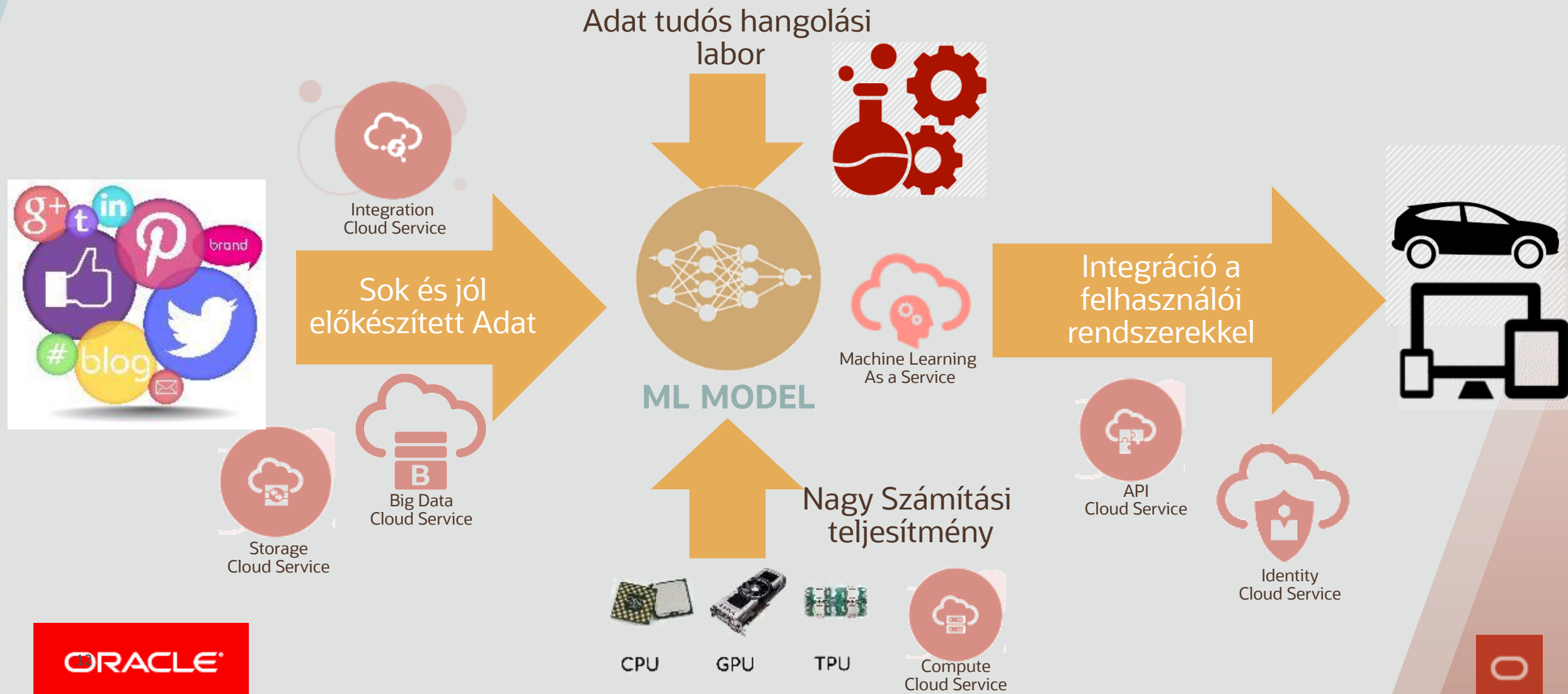
Képes egyetlen gépen vagy elosztott farmokon is futni

Tartalmaz előre elkészített modelleket is

Más keretrendszerek pl. PyTorch kissé eltérő megközelítéssel is rendelkeznek pl. Dinamikus neuron architektúra támogatása



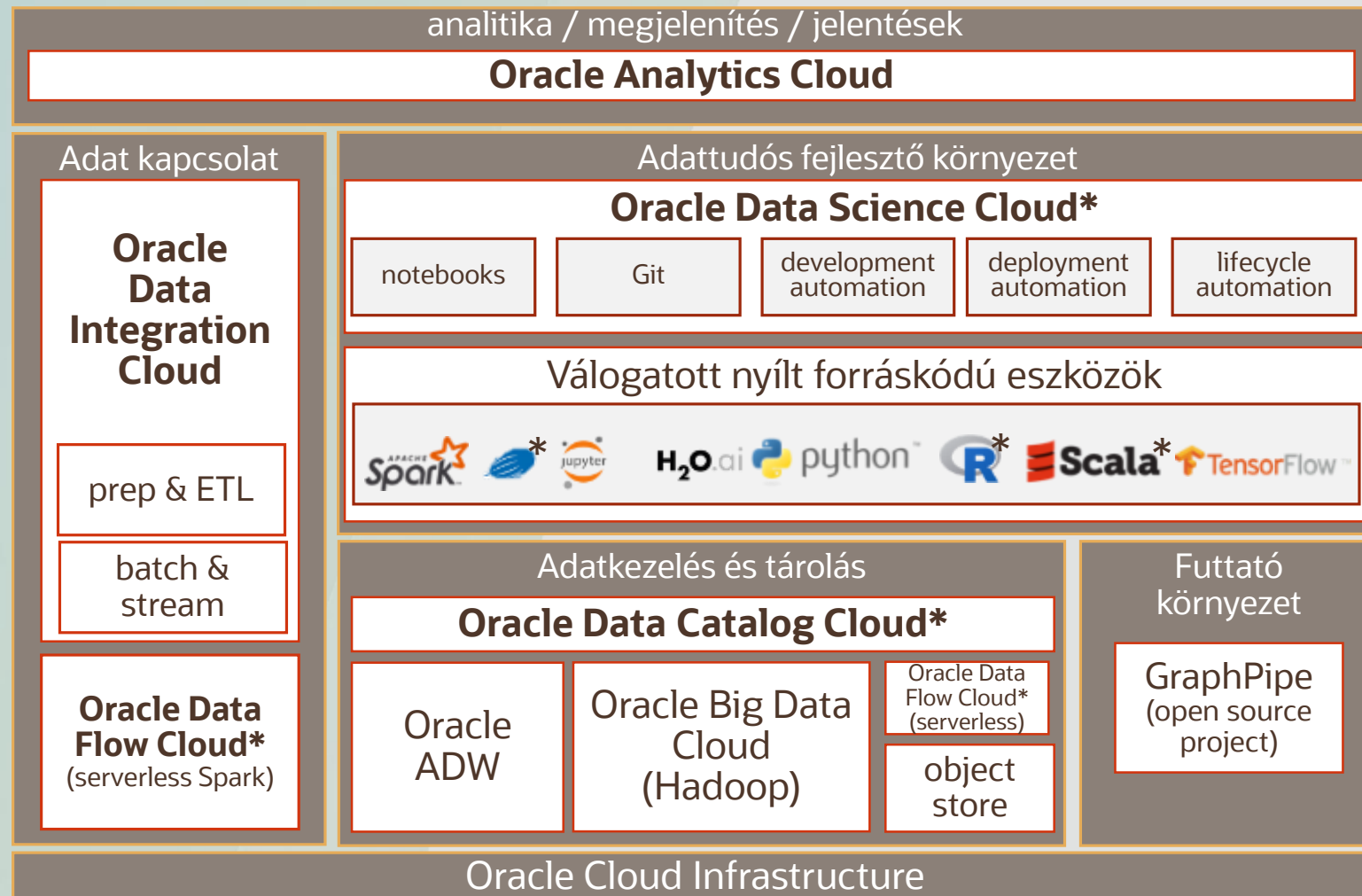
De a Gépi Tanulást klasszikus eszközökkel is támogatni kell



A teljes gépi tanulási életciklus támogatása



A fejlesztői hatékonyság jelentős növelése



Oracle Management Cloud

Oracle Security Cloud

Összefoglalva

A gépi tanulás már nem a scifi könyvek lapjain vagy a TESLA műhelyeiben létezik hanem a vállalati informatikában is

A gépi tanulás hatékony működéséhez, azonban hagyományosabb technológiákból kell köré építenünk adatkezelő, kapcsoló, átalakító és fejlesztő rendszereket

Az Oracle alkalmazásai ma is használnak gépi tanulást és olyan eszközöket is biztosítanak, amikkel mi is fejleszthetünk részeseivé válhatunk a forradalomnak

Köszönöm a Figyelmet

Molnár Balázs

balazs.molnar@oracle.com



APPENDIX

Játszótér: Próbáljunk egy egyszerű feladatot Osztályozás

The screenshot shows a neural network training interface with the following components:

- Top Bar:** Epoch: 000,000; Learning rate: 0.03; Activation: Tanh; Regularization: None; Regularization rate: 0; Problem type: Classification.
- DATA Section:** Includes a 'Which dataset do you want to use?' dropdown, a 'Ratio of training to test data: 50%' slider, 'Noise: 0' slider, and 'Batch size: 10' slider. A 'REGENERATE' button is at the bottom.
- Neural Network Diagram:** Shows an input layer with nodes X_1 and X_2 , a hidden layer with 4 neurons, and an output layer with 2 neurons. Connections are shown with lines of varying thickness representing weights. A tooltip states: 'The outputs are mixed with varying weights, shown by the thickness of the lines.' Another tooltip points to a neuron: 'This is the output from one neuron. Hover to see it larger.'
- OUTPUT Section:** Shows 'Test loss 0.501' and 'Training loss 0.527'. A scatter plot displays data points in blue and orange on a 2D plane. A color scale legend indicates 'Colors shows data, neuron and weight values' with a gradient from 1 (orange) to 0 (blue) to 1 (orange). Checkboxes for 'Show test data' and 'Discretize output' are present.

Annotations:

- Top center: Neuronok rétegekbe rendezve egymással összekapcsolva állnak készen a tanulásra
- Bottom left: Csak két bemenő paraméter ismert az X és Y koordináta
- Bottom center: Pontok színe az ismert tudás, Háttére a kiszámolt tudás
- Bottom right: Ha a háttér és a pont színe azonos, akkor jó a modell
- Right side: Vajon a kék vagy a narancs csoportba tartozik?

Játszótér: Próbáljunk egy egyszerű feladatot Oszályozás

The screenshot shows a neural network training interface with the following elements:

- Epoch:** 000,076
- Learning rate:** 0.03
- Activation:** Tanh
- Regularization rate:** (empty)
- Problem type:** Classification
- DATA:** Which dataset do you want to use? (A dataset icon is selected)
- FEATURES:** Which properties do you want to feed in? (List of features: X_1 , X_2 , X_1^2 , X_2^2 , $X_1 X_2$, $\sin(X_1)$, $\sin(X_2)$)
- 2 HIDDEN LAYERS:** 4 neurons in the first hidden layer, 2 neurons in the second hidden layer.
- OUTPUT:** Test loss 0.032, Training loss 0.028. A scatter plot shows data points clustered in a blue region.
- Callouts:**
 - "Már 76 minta bemutatása után pontos eredményt ér el" (Achieves a perfect result after presenting 76 samples)
 - "A vonalak vastagsága mutatja az egyes kapcsolatok erősségét ezek mátrixban tárolt értéke reprezentálja a tudást" (The thickness of the lines shows the strength of the connections, their values stored in a matrix represent the knowledge)
 - "A modell már 98%-os pontosságú" (The model is already 98% accurate)
 - "Itt láthatjuk milyen gyorsan tanul..." (Here we can see how fast it learns...)
 - "Ez egy viszonylag egyszerű feladat...." (This is a relatively simple task....)

Játszótér: Próbáljunk egy egyszerű feladatot Oszályozás

The screenshot shows a neural network training interface with the following elements:

- Epoch:** 000,044
- Learning rate:** 0.03
- Activation:** ReLU
- Regularization:** None
- Regularization rate:** 0
- Problem type:** Classification

Annotations in Hungarian:

- Es még gyorsabban tanul a hálózat** (And it learns the network even faster) - points to the Epoch counter.
- Változtathatunk néhány hiperparamétert** (We can change some hyperparameters) - points to the Learning rate and Activation dropdowns.
- Vagy pontosabb eredményt produkál** (Or it produces more accurate results) - points to the Test loss and Training loss values.

Network architecture: 2 HIDDEN LAYERS, 2 neurons.

Inputs: X_1 , X_2 , X_1^2 , X_2^2 , X_1X_2 , $\sin(X_1)$, $\sin(X_2)$.

Output: Test loss 0.049, Training loss 0.043.

Legend: Colors show data, neuron and weight values. A color scale from -1 (blue) to 1 (orange) is provided.

Buttons: Show test data, Discretize output.

REGENERATE button is visible at the bottom left.



Játszótér: De adjunk neki egy összetettebb feladatot...

The image shows a neural network simulator interface with several components and annotations:

- Annotations:**
 - Top left: "És még gyorsabban tanul a hálózat" (And it learns even faster).
 - Top center: "Változtathatunk néhány hiperparamétert" (We can change some hyperparameters).
 - Top right: "Vagy pontosabb eredményt produkál" (Or it produces more accurate results).
- Control Panel:**
 - Epoch: 000,044
 - Learning rate: 0.03
 - Activation: ReLU
 - Regularization: None
 - Regularization rate: 0
 - Problem type: Classification
- Network Structure:**
 - Input layer: 2 neurons (X_1, X_2)
 - Hidden layer: 2 neurons
 - Output layer: 2 neurons
- Losses:**
 - Test loss: 0.049
 - Training loss: 0.043
- Visualization:**
 - A scatter plot showing data points (blue and orange) and a decision boundary (blue square).
 - A color scale from -1 (blue) to 1 (orange) for weights and neuron outputs.
 - Buttons: "Show test data", "Discretize output".
- Additional UI Elements:**
 - Left sidebar: "Ratio of training to test data: 50%", "Noise: 0", "Batch size: 10", "REGENERATE".
 - Bottom left: "20".
 - Bottom right: A red square with a white circle.

Játszótér: De adjunk neki egy összetettebb feladatot...

Epoch: 000,000 | Learning rate: 0.03 | Activation: ReLU | Regularization: None | Regularization rate: 0 | Problem type: Classification

DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE

FEATURES
Which properties do you want to feed in?
 X_1
 X_2
 X_1^2
 X_2^2
 $X_1 X_2$
 $\sin(X_1)$
 $\sin(X_2)$

2 HIDDEN LAYERS
4 neurons | 2 neurons
The outputs are mixed with varying weights, shown by the thickness of the lines.
This is the output from one neuron. Hover to see it larger.

OUTPUT
Test loss 0.536
Training loss 0.542
Colors shows data, neuron and weight values.
 Show test data Discretize output

Ezt már azért nehezebb lesz kiszámolni. A feladat ugyanaz, de a minták alapján az eloszlás nem olyan egyszerű



Játszótér: De adjunk neki egy összetettebb feladatot...

Epoch: 000,000
Learning rate: 0.03
Activation: ReLU
Regularization: None
Regularization rate: 0
Problem type: Classification

DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE

FEATURES
Which properties do you want to feed in?
 X_1
 X_2
 X_1^2
 X_2^2
 $X_1 X_2$
 $\sin(X_1)$
 $\sin(X_2)$

2 HIDDEN LAYERS
4 neurons
2 neurons

This is the output from one neuron. Hover to see it larger.

The outputs are mixed with varying weights, shown by the thickness of the lines.

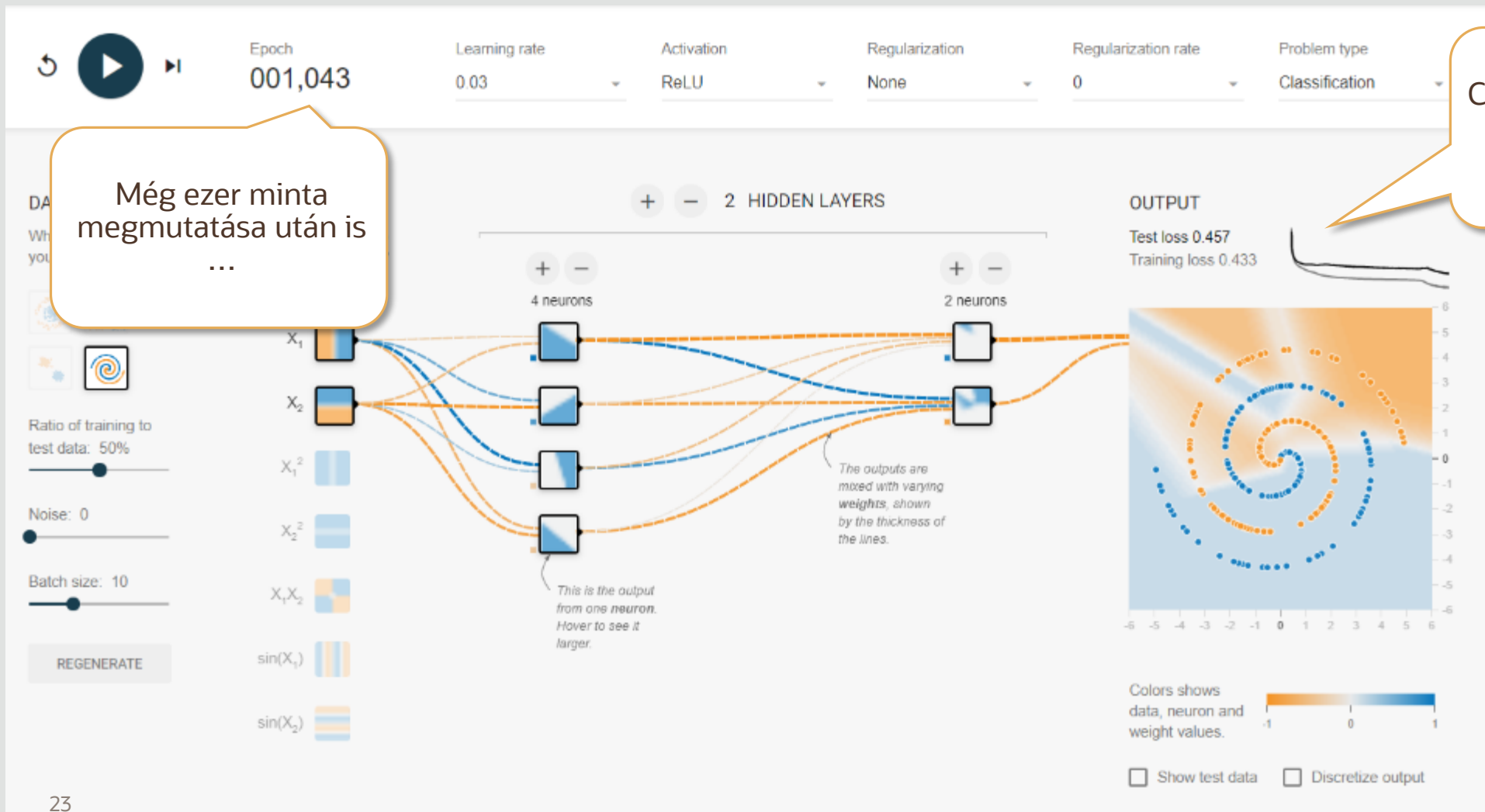
OUTPUT
Test loss 0.536
Training loss 0.542

Colors shows data, neuron and weight values.

Show test data Discretize output

Ezt már azért nehezebb lesz kiszámolni. A feladat ugyanaz, de a minták alapján az eloszlás nem olyan egyszerű

Játszótér: De adjunk neki egy összetettebb feladatot...



Csak 55%-os a modell, valamit változtatni kell...

Még ezer minta megmutatása után is ...



Játszótér: De adjunk neki egy összetettebb feladatot...

Epoch 001,043

Learning rate 0.03

Activation ReLU

Regularization None

Regularization rate 0

Problem type Classification

2 HIDDEN LAYERS

4 neurons

2 neurons

OUTPUT

Test loss 0.457

Training loss 0.433

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

X_1

X_2

X_1^2

X_2^2

$X_1 X_2$

$\sin(X_1)$

$\sin(X_2)$

The outputs are mixed with varying weights, shown by the thickness of the lines.

This is the output from one neuron. Hover to see it larger.

Colors shows data, neuron and weight values.

Show test data Discretize output

Még ezer minta megmutatása után is ...

Csak 55%-os a modell, valamit változtatni kell...



Játszótér: De adjunk neki egy összetettebb feladatot...

Epoch: 000,000
Learning rate: 0.03
Activation: ReLU
Regularization: None
Regularization rate: 0
Problem type: Classification

DATA: Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE

FEATURES: Which properties do you want to feed in?
 X_1
 X_2
 X_1^2
 X_2^2
 $X_1 X_2$
 $\sin(X_1)$
 $\sin(X_2)$

3 HIDDEN LAYERS
8 neurons
8 neurons
8 neurons

OUTPUT: Test loss
Train loss

Colors shows data, neuron and weight values.
-1 0 1
 Show test data Discretize output

Adjunk hozzá még egy réteget és egy réteget és nézzük meg, így gyorsabb lesz-e?



Játszótér: De adjunk neki egy összetettebb feladatot...

