MLP az MNIST adatkészletre

Projekt Áttekintés

Ez a projekt egy **többrétegű perceptron (MLP)** neurális hálót és egy hozzá tartozó **interaktív grafikus felhasználói felületet (GUI)** tartalmaz.
A rendszer elsődleges célja a **kézzel rajzolt számok felismerése (0–9)**, MNIST-szerű képek alapján.

A felhasználó képes:

- szabadkézzel rajzolni egy számot nagy méretű, pixeles canvason,
- a neurális háló azonnal megjósolja, hogy milyen szám lett rajzolva,
- a háló működése közben vizualizálni a háló rejtett rétegeinek aktivációit,
- ha a modell téved, a felhasználó megadhatja a helyes címkét,
- a modell ezek után **valós időben tanul** (online learning),
- a model továbbá betanítható MNIST-ről, illetve elmenthető és betölthető.

A projekt részei:

- **mlp_mnist.py** → az MLP modell + MNIST betöltő
- **gui_mlp.py** → a teljes grafikus felület, vizualizáció, model kezelés
- **model.npz** → elmentett súlyok (opcionális)

A Megoldás Fő Funkciói

1. **Többrétegű perceptron háló** saját NumPy implementációval
    - ReLU aktiváció rejtett rétegeken
    - Softmax aktiváció kimeneti rétegen
    - Cross-entropy veszteség
    - Hátraterjesztés (backpropagation) minden réteghez
    - Batch-alapú tanítás és online tanulás
2. **GPU-független, NumPy-alapú működés**
3. **MNIST betöltése TensorFlow nélkül**
4. **Interaktív GUI**
    - Nagy rajztábla, 28×28 mintavételezéssel
    - Predikciók és valószínűségek megjelenítése
    - Háló topológiájának vizuális kirajzolása
        - max. 50 neuron/réteg
        - aktiváció-alapú színezés
        - súlyok vizuális megjelenítése
    - Modell mentése / betöltése
    - Gyors MNIST tanítás

- Online tanulás (felhasználó visszajelzése alapján)

A MLP Modell Felépítése

A neurális háló egy rugalmas, többrétegű MLP:

Alap architektúra (ajánlott):

- Input réteg: **784** (28×28 pixel)
- Rejtett réteg 1: **128 neuron**, ReLU
- Rejtett réteg 2: **64 neuron**, ReLU
- Kimeneti réteg: **10 neuron**, Softmax

A Modell Működése

Előrehaladás (Forward Pass)

1. A bemeneti vektor (784 hosszú) sorban áthalad minden DenseLayer-en.
2. Minden réteg elvégzi:
   - Z = A_prev @ W + b
   - A = activation(Z)
3. A végső kimenet egy 10 elemű softmax vektor:
   - [p0, p1, ..., p9]

Tanítás

Kétféle tanítási mód működik:

(A) Batch training – MNIST betöltésével

- Veszteség: cross-entropy
- Gradiens: dZ = A_L - Y_one_hot
- Súlyfrissítés: W -= lr * dW

(B) Online training – GUI-ban kézzel

A felhasználó megadja, mi a helyes szám → 1 lépés tanítás történik.

A GUI felépítése

A GUI-t a gui_mlp.py tartalmazza.
Fő komponensek:

Rajztábla (Canvas)

- Mérete: 560×560 px
- Logikai felbontás: 28×28
- Minden cella 20×20 pixel

- Egérhúzás → belső 28×28 grid frissítése

Predikció és valószínűségek panel

- Megjeleníti a pillanatnyi előrejelzés eredményét
- Kiírja a softmax valószínűségeket

Interaktív tanítás (online learning)

- Ha a predikció rossz:
    - kattintasz a helyes számra
    - a háló egy lépést tanul a jelenlegi mintából

Modellkezelő gombok

- Modell mentése → model.npz
- Modell betöltése → model.npz
- Gyors MNIST tanítás (20k minta, 3 epoch)

Háló vizualizáció

A hálózat aktivációit vizuálisan jeleníti meg:

Neuronok:

- pontok (körök)
- színük a neuron aktiváció értéke alapján:
    - sötét → alacsony aktiváció
    - világos → magas aktiváció

Súlyok:

- vonalak a rétegek között
- szín:
    - piros → pozitív súly
    - kék → negatív súly
- intenzitás → súly nagysága

Megjelenítési korlát:

Maximum 50 neuron/réteg → ez biztosítja, hogy:

- gyors maradjon a frissítés
- áttekinthető legyen a grafika

Modell Mentése és Betöltése

Mentés

A súlyok NumPy tömörített .npz formátumban kerülnek a fájlba:

- layer0_W, layer0_b
- layer1_W, layer1_b
- layer2_W, layer2_b
- ...

```
save_model(model, "model.npz")
```

Betöltés

A súlyok visszatöltése:

```
model = load_model("model.npz")
```

Online Tanulás a GUI-ban

Ha a felhasználó rajzolt egy számot, a modell prediktál.
Ha hibás → a felhasználó rákattint a helyes szám gombra.

Ekkor:

```
self.model.fit_batch(x, np.array([correct_digit]), lr)
```

Ez egyetlen mintán végrehajtott tanítást jelent.
A tanulás azonnal érzékelhető a háló vizualizációban és a következő predikciókban.

Háló Vizuális Megjelenítése

A GUI a háló aktivációit és súlyait a jobb oldali canvas-on rajzolja ki.

Réteg elrendezés

- vízszintesen a rétegek egymás mellett
- függőlegesen: kiválasztott neuronok elosztva

Neuron aktiváció színezés

```
(minA → sötét) ... (maxA → világos)
```

Súly színezés

Súly érték	Szín	Jelentése
pozitív	piros	serkentő kapcsolat
negatív	kék	gátló kapcsolat
0 közeli	szürke	gyenge kapcsolat

Továbbfejlesztési Ötletek

- dropout implementálása
- tanulási ráta beállítása GUI-ból
- új vizualizációs nézet (pl. hőtérkép a súlyokra)
- konvolúciós réteg hozzáadása
- több rejtett réteg megjelenítése külön tabokban
- modell statisztika (loss görbe) hozzáadása

Források:

https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

https://www.youtube.com/watch?v=SmZmBKc7Lrs

https://www.kaggle.com/code/manzoormahmood/mnist-neural-network-from-scratch/notebook

https://www.youtube.com/watch?v=w8yWXqWQYmU

https://medium.com/@ombaval/building-a-simple-neural-network-from-scratch-for-mnist-digit-recognition-without-using-7005a7733418

https://www.youtube.com/watch?v=Ixl3nykKG9M

Érdekes matematikai/programozási videók és videósorozatok:

https://www.youtube.com/@3blue1brown/featured