

# Neurális Hálózat Toolboxok/Libraryk

AIDS Csapat

PPKE-ITK Python Technológiák

2016. március 9.

## Miről is lesz szó?

- ▶ Rövid elméleti áttekintés (felügyelt tanulás, neurális hálózatok).
- ▶ Csomagok (Theano, Caffe, Lasagne) bemutatása.
- ▶ Példa.

## Elérhetőség

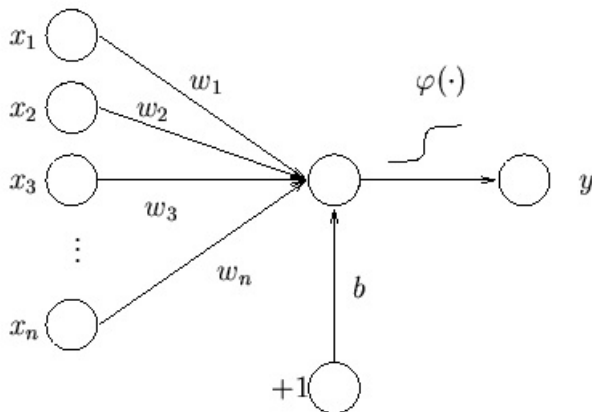
A prezentáció és a hozzá tartozó anyagok elérhetőek:  
<http://bit.ly/1TKa5z2>.

A felügyelt tanulási folyamatokban előre felcímkezett adatok segítségével egy tanulási fázis után a rendszer képessé válik még ismeretlen adatok (*jobb esetben helyes*) klasszifikációjára. Ehhez szükséges egy tanítóhalmaz (1).

$$T = \{(\bar{x}, d)\} \tag{1}$$

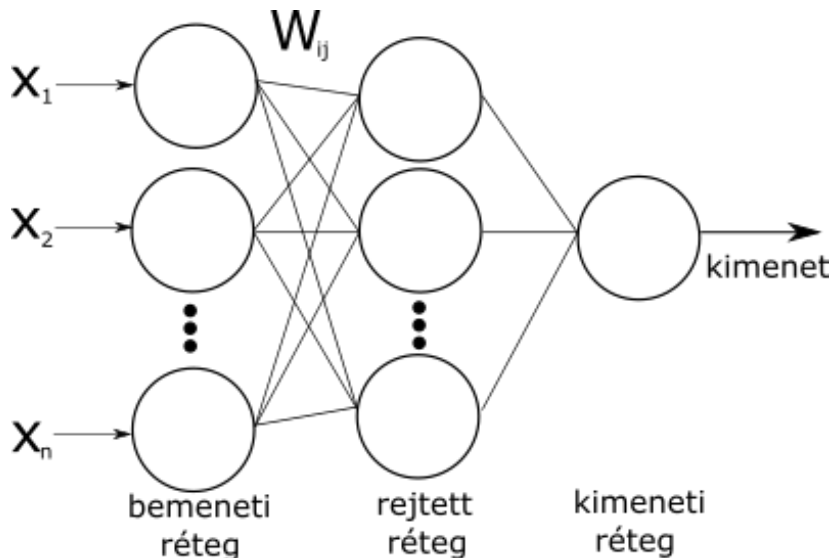
# Gépi Tanulás - Neurális Hálózatok

A neurális hálózatok használatának előnye, hogy a rendszer saját maga nyeri ki a tanító mintákból az azokat leíró tulajdonságokat, ún. *feature*-öket.



# Gépi Tanulás - Neurális Hálózatok

A NN-ek alapegysége a mesterséges neuron, ezeket hálózatba kötve egy ún. Feedforward NN-t kaphatunk.



# Neurális Hálózatok - Backpropagation

- ▶ A tanulás az ún. backpropagation algoritmuson keresztül valósul meg. Amennyiben eltér az elvárt kimenet a rendszer aktuális kiemenetétől, a súlyokat módosítja.
- ▶ Mivel a hálózat állapota a súlyokban tárolódik, a backpropagation algoritmus a rendszer állapotátmenet-függvénye.
- ▶ Az algoritmus segítségével egy gradiens mentén minimalizáljuk a hibát. A működése viszont nem tökéletes: képes beragadni lokális minimumokba és a gradiens felrobbanhat/eltűnhet. *(És egyéb ML rákfénék...)*

Convolutional Architecture for Fast Feature Embedding:

- ▶ C++ -ban íródott, CUDA + cuDNN támogatás.
- ▶ Python és MATLAB bindingokkal.
- ▶ CLI.
- ▶ Főként konvolúciós hálózatokra készült.

Elérhető: <http://caffe.berkeleyvision.org/>

# CAFFE - Bemeneti Adatok

- ▶ Preferáltak a kulcs-érték pár alapú DB-k (LMDB, LevelDB), de kezeli:
- ▶ HDF5,
- ▶ átlagos képformátumok pl. JPEG.

A hálózatokat Google Protocol Buffer file-okban tárolja:

```
layer {  
  name: "loss"  
  type: "SoftmaxWithLoss"  
  bottom: "score"  
  bottom: "label"  
  top: "loss"  
}
```



# CAFFE - Példák

A PyCAFFE működésére példa:

`http://nbviewer.jupyter.org/github/BVLC/caffe/blob/master/examples/01-learning\_lenet.ipynb`

## Megjegyzések

- ▶ A *solver* fájl tartalmazza a tanítás paramétereit (iterációk száma, snapshotok gyakorisága). Ezt külön fájlban adjuk meg.
- ▶ A hálózat leírása is megadható kézzel külön fájlban, Google ProtoBuf formátumban.
- ▶ A CAFFE nem kezeli az egyes rétegek részeként a nemlinearitásokat, azokat a rétegek után kell tenni.

## Lasagne - Theano

A Lasagne működése a Theano könyvtárra épül, mely matematikai kifejezések optimalizálására készült, főként többdimenziós tömböket tartalmazóakéra. (Mint esetünkben a súlymátrixok.)

```
import theano
from theano import tensor
a = tensor.dscalar()
b = tensor.dscalar()
c = a + b
f = theano.function([a,b], c)
assert 4.0 == f(1.5, 2.5)
```

## Bemeneti Adatok

- ▶ Szinte bármit megeshzik, amit NumPy arrayekbe tudunk tenni.
- ▶ Könnyen integrálható pipeline-okba.

## megjegyzések

- ▶ A Caffe-val ellentétben az egyes rétegek részeként kezeli a nemlinearitásokat (már ahol van értelme).
- ▶ Nagyságrendekkel jobban dokumentált (Docstringek, Tutorialok), mint a Caffe.

# Lasagne - Példák

A Theano elérhető:

<http://deeplearning.net/software/theano/>

Theano tutorial és példakódok:

<http://deeplearning.net/software/theano/tutorial/>

A Lasagne maga pedig elérhető itt:

<http://lasagne.readthedocs.org/en/latest/index.html>

Az előbb látott MNIST-példa Lasagne megfelelője:

<https://github.com/Lasagne/Lasagne/blob/master/examples/mnist.py>