

Kínai karakterek felismerése konvolúciós neurális hálók használatával

Szilvási Péter

Miskolci Egyetem, 2019. január 18.

Kínai karakterek felépítése

永

- Dokumentumok digitalizálása

- Feldolgozási szintek: [2]

Alacsony szintű

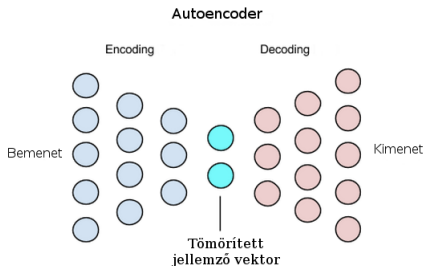
Középső szintű

Magas szintű



Jellemzők kinyerése, Dimenzió redukció

- Főkomponens analízis (*PCA*)
 - Magas dimenzió \rightarrow Alsó dimenzió
 - $Av = \lambda v$
- Kernelek alkalmazása
 - Nem lineáris leképezések
- Neurális háló szerkezete



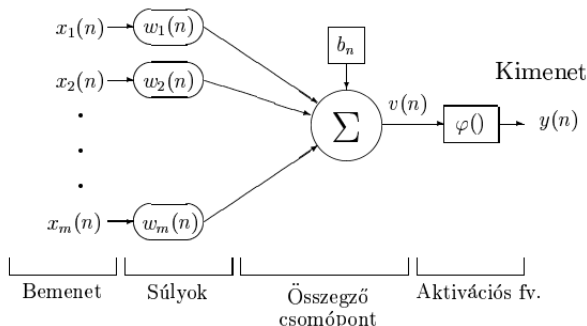
Mesterséges neurális hálók

Neurális hálózatok [5]

- Rétegek
- Elemei

Backpropagation

- Hiba $E_{total} = \sum \frac{1}{2} (target - output)^2$.
- Láncszabály $\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \cdot \frac{\partial out_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial w_5}$.



Konvolúciós neurális háló

- Hálózat felépítése (konvolúciós rétegek → hagyományos ANN)
- Bemenet → (*Konvolúció* → *RELU* → *POOL*) → Kimenet(FC)

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

Input x Filter

4		

Feature Map

Single depth slice

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

x

y

max pool with 2x2 filters
and stride 2

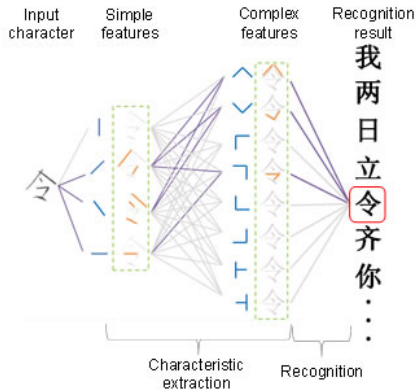
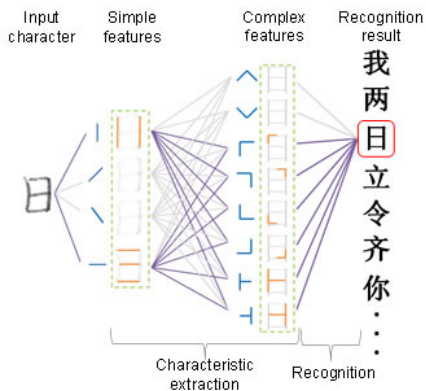
6	8
3	4

- Hálózat tanítás

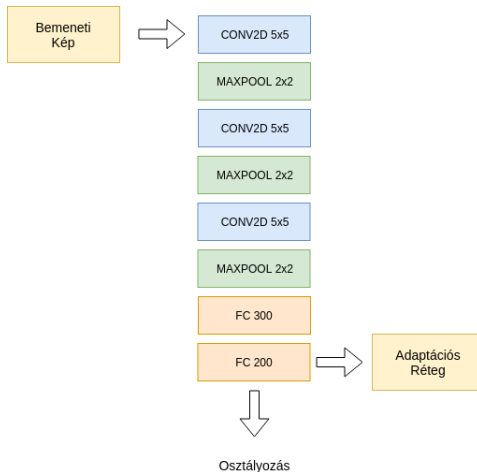
1. Előre terjesztés
2. Veszteség számítás
3. Hiba visszaterjesztés
4. Súly frissítés

- Dropout

Felismerés



A hálózat architektúrája



Előfeldolgozás

Adathalmaz: nyomtatott, kézzel írott, generált

高 九 東 且 世

- Kép manipulálás:

Átméretezés

Forgatás

Zajosítás

Vágás

```
def resize():    def rotate():    def noise():
```

- Konvertálás:

```
for myFile in files:
    image = cv2.imread (myFile)
    train.append (image)
    train_labels.append([1., 0., 0., 0., 0.])

np.reshape() -> np.save()
```

Futtatás

```
time python3 train.py | tee log
```

```
Epoch 1/80
```

```
12/2000 [.....] loss:0.6964 acc: 0.7833
```

```
24/2000 [.....] loss:0.7951 acc: 0.7833
```

```
36/2000 [.....] loss:1.0486 acc: 0.7278
```

```
Epoch 80/80
```

```
1992/2000 [=====>.] loss:0.0385 acc:0.9871
```

```
2000/2000 [=====] loss:0.0383 acc:0.9872
```

```
real    19m40,397s
```

```
user    55m4,422s
```

```
sys      5m41,888s
```

Tesztelés

- Kimenet létrehozás (.csv)
- Model beolvasás

```
model = load_model('model1.h5')
```

- Kép átalakítás

```
x = x.reshape((28,28,3))
```

- Megjósolás

```
x = model.predict(x)
```

- Eredmény -> Kimeneti fájl

- 1 Tikk Domonkos: *Optikai karakterfelismerés*, online melléklet, TypoTeX kiadó, 2006.
- 2 Rövid A., Vámosy Z., Sergyán S.: *A gépi látás és képfeldolgozás párhuzamos modelljei és algoritmusai*, 2014.
- 3 Liu, Yin, Wang, Wang: *Online and offline handwritten chinese character recognition: benchmarking on new databases*, Pattern Recognition, 2013.
- 4 X. Wu, M. Wu: *A recognition algorithm for chinese characters in diverse fonts*, Image Processing, 2002.
- 5 Fazekas István: *Neurális hálózatok*, Debreceni Egyetem, 2013.

Köszönöm szépen a figyelmet!