

# Kínai karakterek felismerése konvolúciós neurális hálók használatával

Szilvási Péter

Miskolci Egyetem, 2019. január 24.

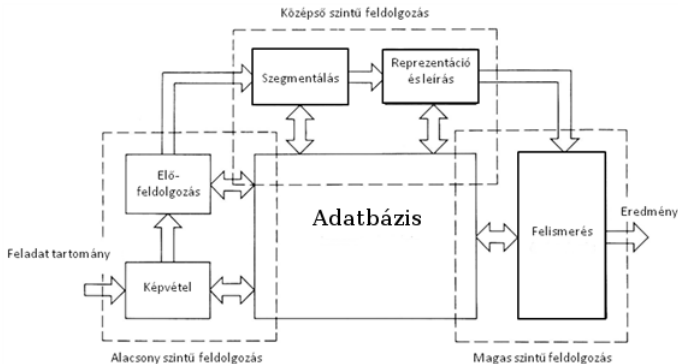


## Vonások, vonás sorrend

- 1 A vízszintes vonások megelőzik a függőleges vonásokat.
- 2 A balra lejtő vonások megelőzik a jobbra lejtő vonásokat.
- 3 Az írásjegyek írását felülről kell kezdeni.
- 4 Az írásjegyet balról jobbra haladva építik fel.
- 5 A felülről keretezett írásjegyeknél előbb a keretet kell meghúzni.
- 6 Az alulról keretezett írásjegyeknél a keretet legvégül kell meghúzni.
- 7 A teljes keretet mindig legvégül kell bezárni.

# OCR megvalósítások

- Dokumentumok digitalizálása
- OCR részei: szkennelő fej + szoftver [1]
- Feldolgozási szintek: [2]
  - Alacsony szintű: zajos kép → előfeldolgozás → javított kép
  - Középső szintű: kép → szegmentálás → kép jellemzők
  - Magas szintű: jellemzők → osztályozás → osztálycímke

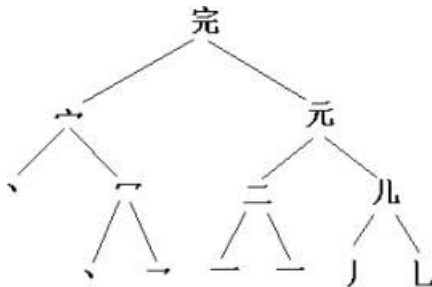


# OCR megvalósítások

OCR típusok: *online*, *offline* [3]

Kínai karakter felismerése

- Zaj szűrés: pontszerű zajok, elmosódás, forgatás, kontraszt
- Jellemzők kinyerése



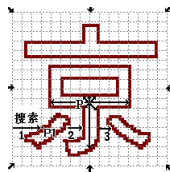
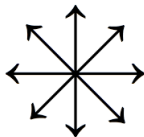
# OCR megvalósítások

|      |   |   |   |   |
|------|---|---|---|---|
| Song | 多 | 体 | 汉 | 字 |
| Fang | 多 | 体 | 汉 | 字 |
| Kai  | 多 | 体 | 汉 | 字 |
| Hei  | 多 | 体 | 汉 | 字 |

## Egy elterjedt algoritmus [4]

### ■ Dimenzió redukció

$$d_i = \frac{l_i}{\sqrt{\sum_{k=1}^8 l_k^2}}$$



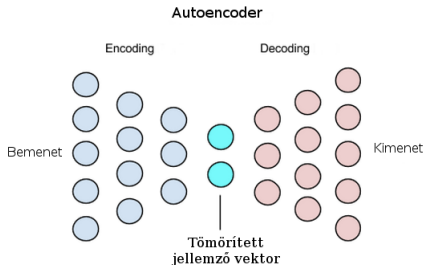
### ■ Tanítás

### ■ Tesztelés

| Font  | Song  | Fang  | Kai   | Hei   |
|-------|-------|-------|-------|-------|
| Train | 99.82 | 99.64 | 99.81 | 99.57 |
| Test  | 99.71 | 99.50 | 99.80 | 99.09 |

# Jellemzők kinyerése, Dimenzió redukció

- Főkomponens analízis (*Principle Component Analysis*)
  - Magas dimenzió → Alsó dimenzió
  - $Av = \lambda v$
- Kernelek alkalmazása
  - Nem lineáris leképezések
- Neurális háló szerkezete



# Alacsony szintű jellemzők

- Éldetektálás
- Sarokérzékelés
- Skála invariáns jellemző transzformáció (SIFT, *Scale Invariant Feature Transform*)

# Írány szerinti jellemző kinyerés

- Irány dekompozíció
- Elmosódás és mintavétel
- Számítási idők
- Korlátozás nélküli minták



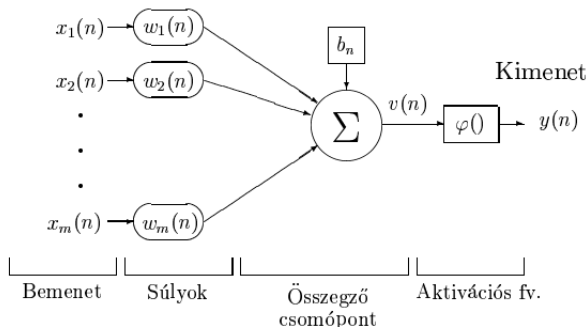
# Mesterséges neurális hálók

## 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  $\rightarrow$  hagyományos ANN)
- Bemenet  $\rightarrow$  (Konvolúció  $\rightarrow$  RELU  $\rightarrow$  POOL)  $\rightarrow$  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

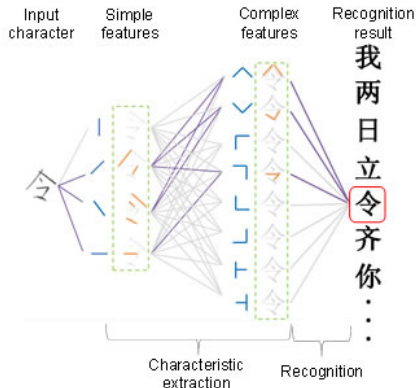
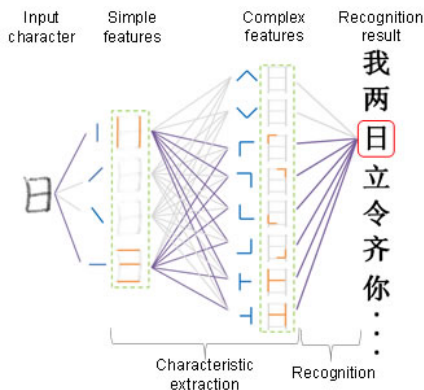
|   |   |
|---|---|
| 6 | 8 |
| 3 | 4 |

- Hálózat tanítás

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

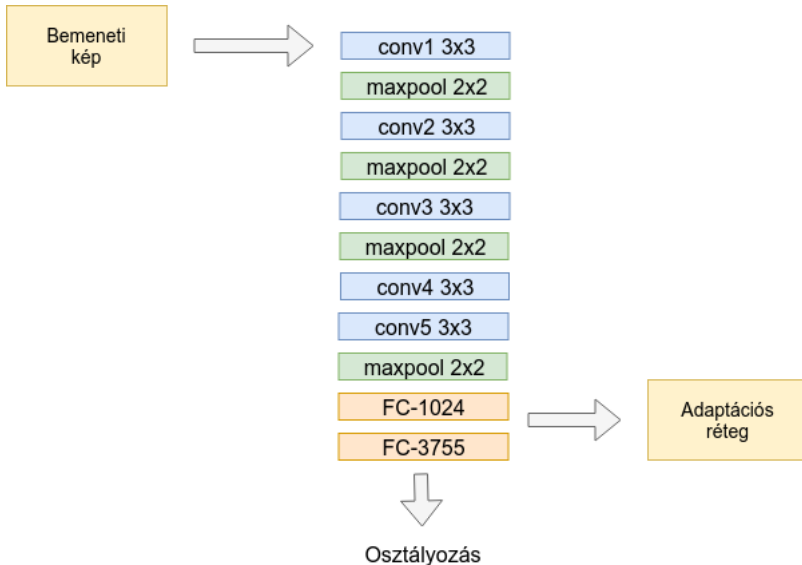
- Dropout

# A háló felépítése



- Tesztelés
- Transfer learning

# A hálózat architektúrája



# A hálózat architektúrája

```
model.add(
    Convolution2D(1,          # filter retegek szama
                  3, 3,       # 3x3 kernel meret
                  strides=(1,1) # lepes
                  input_shape=image))
```

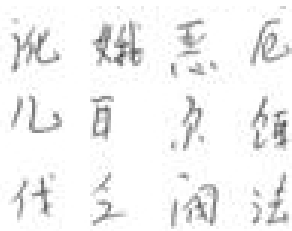
```
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.compile(loss='mean_squared_error', # Hiba
              optimizer='adam', metrics=['accuracy'])
```

```
model.fit_generator(generator=training_data,
                    steps_per_epoch=1000, epochs=10)          # Tanitas
```





# Az offline adatbázis

Adathalmaz: nyomtatott, kézzel írott, generált



- Tanító/Teszt(80/20), `random.shuffle(self.images)`
- Tanító minták változatossága
- Tesztelés módja
- Helyesség ellenőrzése

# A felismerés hatékonysága

|        |   |   |   |        |
|--------|---|---|---|--------|
| 0%     | → |  | → | 90-95% |
| 75-80% | → |  | → | 84%    |
| 68%    | → |  | → | 75%    |
| 87%    | → |  | → | 93%    |

# Összegzés

- Kínai karakterek
  - stroke
  - vonásrend
- OCR
  - részei
  - használt OCR bemutatás
- Jellemzők kinyerése
  - dimenzió redukció
  - alacsony szintű jellemzők
  - irány szerinti jellemző kinyerés
- Neurális hálózatok
  - hagyományos neurális háló (ANN)
  - konvolúciós neurális háló (CNN)
- Validáció
  - adathalmaz előállítás
  - hálózat osztályozása



- 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!**