Number classification on Mnist database also optimizing

Feladat:

A feladat, hogy egy 0-tól 9-ig való számokból álló adatbázison végezzünk classificationt. Ehhez az Mnist adatbázist használom. Itt próbálkoztam kisebb optimalizálással amiket később felhasználtam már, earlystopping dropout továbbá visszatöltöttem a legjobb modellt a végén, kicsit használtam a seaborn és megnéztem a konfúziós mátrixot is.

A modell felépítése:

1. Adatok betöltése
2. Alapmodell felépítése
3. EarlyStopping + Best model
4. Training
5. Best model visszatöltése
6. Konfúziós mátrix + seaborn

A modell felépítése:

```
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 24, 24, 32)        832
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 20, 20, 64)        51264
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 10, 10, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 10, 10, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 6400)              0
_________________________________________________________________
dense (Dense)                (None, 128)               819328
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290
=================================================================
Total params: 872,714
Trainable params: 872,714
Non-trainable params: 0
_________________________________________________________________
None
```

A dropoutról más meséltem a data augmentációs dokumentációban, ebben a munkában használtam először ilyet, ebben a modellben mondjuk nem annyira fontos a dropout mint a másikban, ott többet segít. A flattening azért fontos, hogy az adatokból 1ds adatokat csináljunk, mert csak azt tudjuk a dense rétegnek átadni. A képeket 28X28-as képekké alakítom. Az első modellben próbálkoztam, csak dense régeteggel de az nagyon rossz eredményeket hozott, a dropoutot egy videóban láttam és azért tettem bele, de itt szinte mindegy a szerepe , amúgy elég hasznos tud lenni az augmentációs modellemben segített valamennyit.

Optimizernek az adamot használtam, észrevételeim alapján szinte mindig az a legjobb, kipróbáltam még egyet kettőt de semelyik sem adott olyan jó eredményeket bár azért volt ami megközelítette.

Egy teszt futtatása:

```
Epoch 1/30
  1/375 [..............................] - ETA: 0s - loss: 2.3284 - accuracy: 0.0703WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/summary_ops_v2.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
375/375 [==============================] - ETA: 0s - loss: 0.2497 - accuracy: 0.9248
Epoch 00001: val_loss improved from inf to 0.05527, saving model to model.hdf5
375/375 [==============================] - 162s 431ms/step - loss: 0.2497 - accuracy: 0.9248 - val_loss: 0.0553 - val_accuracy: 0.9826
Epoch 2/30
375/375 [==============================] - ETA: 0s - loss: 0.0814 - accuracy: 0.9755
Epoch 00002: val_loss improved from 0.05527 to 0.04200, saving model to model.hdf5
375/375 [==============================] - 162s 432ms/step - loss: 0.0814 - accuracy: 0.9755 - val_loss: 0.0420 - val_accuracy: 0.9875
Epoch 3/30
  7/375 [..............................] - ETA: 2:08 - loss: 0.0688 - accuracy: 0.9833
```

A modell nagyon jól működik, szinte 100%-os a tanítási ráta. Nem futtattam túl sokáig, mert nagyon sok időbe telt volna.

A teszthet tartozó legjobb modell visszatöltve és lefuttatva:

```
313/313 [==============================] - 9s 30ms/step - loss: 0.0373 - accuracy: 0.9873
Teszt hiba: 0.03729455545544624 Teszt pontosság: 0.9872999787330627
```

Egy másik teszt végig futtatva, dropout rétegek nélkül:

```
Epoch 1/10
960/960 [==============================] - ETA: 0s - loss: 0.0476 - accuracy: 0.9850
Epoch 00001: val_loss improved from 0.05950 to 0.04928, saving model to model.hdf5
960/960 [==============================] - 171s 179ms/step - loss: 0.0476 - accuracy: 0.9850 - val_loss: 0.0493 - val_accuracy: 0.9861
Epoch 2/10
960/960 [==============================] - ETA: 0s - loss: 0.0275 - accuracy: 0.9911
Epoch 00002: val_loss did not improve from 0.04928
960/960 [==============================] - 170s 178ms/step - loss: 0.0275 - accuracy: 0.9911 - val_loss: 0.0625 - val_accuracy: 0.9827
Epoch 3/10
960/960 [==============================] - ETA: 0s - loss: 0.0184 - accuracy: 0.9941
Epoch 00003: val_loss improved from 0.04928 to 0.04014, saving model to model.hdf5
960/960 [==============================] - 167s 174ms/step - loss: 0.0184 - accuracy: 0.9941 - val_loss: 0.0401 - val_accuracy: 0.9886
Epoch 4/10
960/960 [==============================] - ETA: 0s - loss: 0.0130 - accuracy: 0.9956
Epoch 00004: val_loss did not improve from 0.04014
960/960 [==============================] - 167s 174ms/step - loss: 0.0130 - accuracy: 0.9956 - val_loss: 0.0502 - val_accuracy: 0.9865
Epoch 5/10
960/960 [==============================] - ETA: 0s - loss: 0.0111 - accuracy: 0.9963
Epoch 00005: val_loss did not improve from 0.04014
960/960 [==============================] - 164s 171ms/step - loss: 0.0111 - accuracy: 0.9963 - val_loss: 0.0425 - val_accuracy: 0.9900
Epoch 6/10
960/960 [==============================] - ETA: 0s - loss: 0.0089 - accuracy: 0.9971
Epoch 00006: val_loss did not improve from 0.04014
960/960 [==============================] - 165s 172ms/step - loss: 0.0089 - accuracy: 0.9971 - val_loss: 0.0437 - val_accuracy: 0.9893
Epoch 7/10
960/960 [==============================] - ETA: 0s - loss: 0.0073 - accuracy: 0.9975
Epoch 00007: val_loss did not improve from 0.04014
960/960 [==============================] - 167s 174ms/step - loss: 0.0073 - accuracy: 0.9975 - val_loss: 0.0460 - val_accuracy: 0.9893
Epoch 00007: early stopping
```

A futást megállította az early stopping, mert 4 epochon keresztül nem tudott a modell javulni.

A teszthez tartozó konfúzis mátrix továbbá a test accuracy és egyéb tulajdonságok. Az f1 score érdekes, mert ezt a precisionből és a recallból kalkulálják és elméletileg egy írja le az egyik legjobban a modellt, ennek ha jól olvastam így számítják ki az eredményét:

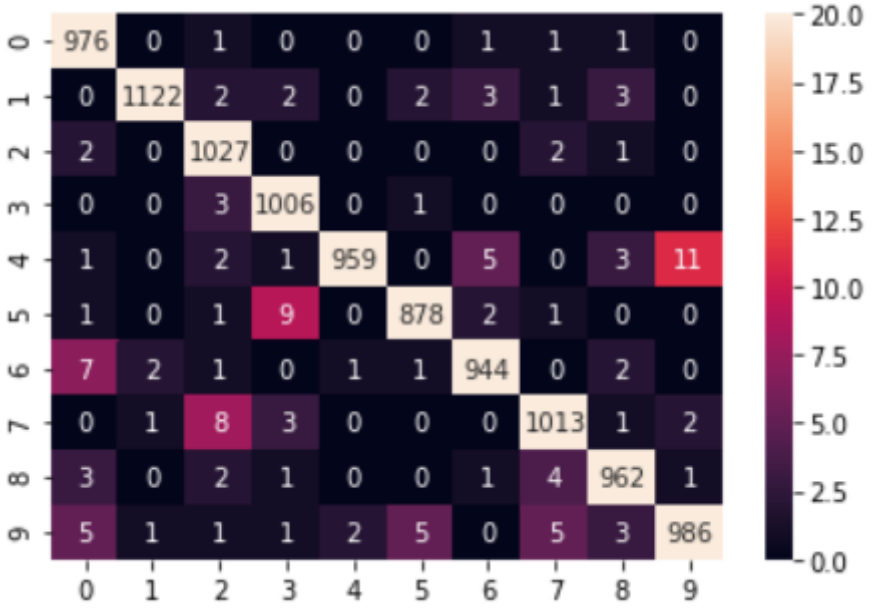
$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}.$$

Konfúziós mátrix és test adatok:

```
test accuracy: 0.9873
Precision 0.9872902600199783
Recall 0.987222221639535
f1_score 0.9872256478534117

Konfúziós mátrix:
[[ 976    0    1    0    0    0    1    1    1    0]
 [   0 1122    2    2    0    2    3    1    3    0]
 [   2    0 1027    0    0    0    0    2    1    0]
 [   0    0    3 1006    0    1    0    0    0    0]
 [   1    0    2    1  959    0    5    0    3   11]
 [   1    0    1    9    0  878    2    1    0    0]
 [   7    2    1    0    1    1  944    0    2    0]
 [   0    1    8    3    0    0    0 1013    1    2]
 [   3    0    2    1    0    0    1    4  962    1]
 [   5    1    1    1    2    5    0    5    3  986]]
```

A végére maradt, a seaborn amivel meg tudjuk jeleníteni, hogy a modell miket tippelt félre és miket talált el.



Az átlóban található az amikor a modell azt tippelte ami a szám. Y tengelyes a tényleges szám míg x tengelyen a tipp látható és az adott x,y-hoz tartozó szám pedig hogy mennyire történt ez meg. Látszik, hogy a modell legjobban a 4-est keverte fel a 9-sel, ennek valószínűsíthetően az az oka, hogy mind a kettő és zárt „köralakú” dologgal indul a tetején (4-nél ez nem annyira kör de hasonlít) majd lefele van egy kis szár. Ahogy néztem érdekes dolgokat néznek meg ezek konvolúciós rétegek, pont valami hasonlót, amit leírtam nagyon nagy vonalakban.