Neurónové siete – projekt Klasifikácia cifier za pomoci EEG dát

(riešenie problému klasifikácie/predikcie pomocou hlbokých neurónových sietí)

Riešiteľ: **Marek Dorko, 1Im**, Ústav informatiky PF UPJŠ

Rok: 2022/2023, zimný semester

Formulácia problému

Úlohou projektu je vytvoriť klasifikátor cifier, ktorý skúma súbor/dataset – v našom prípade EEG dáta a z hodnôt, ktoré boli namerané v priebehu dvoch sekúnd pre každú cifru klasifikovať, o ktorú cifru sa jedná.

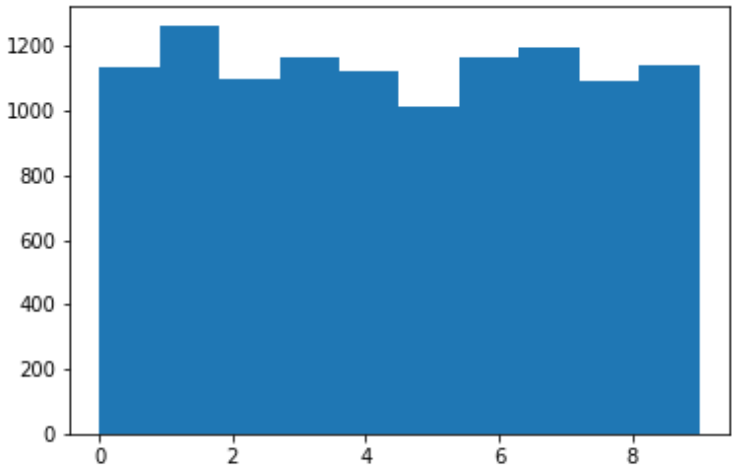
Príprava dát

Dataset sme získali z webovej stránky MindBigData, kde sa nachádza viacero súborov. Rozhodol som sa pre dataset **Muse2-v0.16Cut2**, teda pre súbor *MindBigDataVisualMnist2021-Muse2v0.16Cut2.csv*

Súbor obsahuje 11 386 cifier (jeden riadok jedna cifra), pričom každý riadok v datasete predstavuje jedno EEG meranie. Formát súboru je nasledovný:

- text, ktorý hovorí o type, môže byť TRAIN/TEST,
- celé číslo – v našom prípade nepodstatné,
- celé číslo – pôvodná cifra zobrazená na obrázku, môže byť 0-9, prípadne -1, ak nie je zobrazená žiadna číslica,
- 784 celých čísel s pôvodnými intenzitami pixelov zo zobrazených (28x28) png obrázkov, každý pixel môže mať hodnotu 0-255,
- timestamp,
- EEG dáta – 512 * 2 hodnôt, v tomto prípade 2 kanály TP9 a TP10 – raw signal (2secs at 256hz),
- ďalšie hodnoty sú pre nás nepodstatné

Početnosti jednotlivých cifier v datasete znázorňuje nasledujúci histogram:

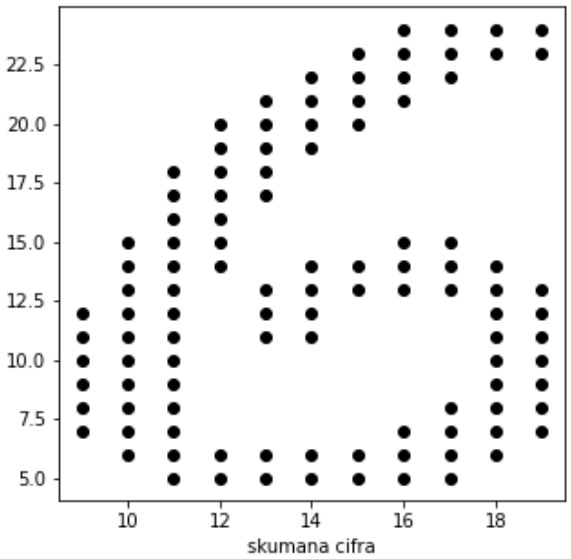


cifra	počet
0	1135
1	1236
2	1096
3	1167
4	1120
5	1012
6	1164
7	1196
8	1091
9	1142

Obr. 1 Histogram početností cifier v datasete

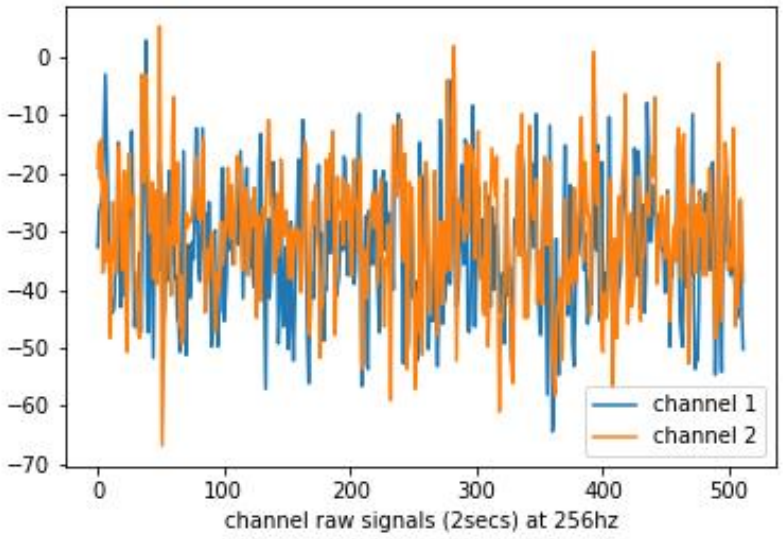
Predspracovanie dát

V úvode sme spracovali dáta, ktoré reprezentujú png obrázok veľkosti 28x28 (784 hodnôt). Napríklad pre cifru ‘6’ je výstupom nasledujúci obrázok:



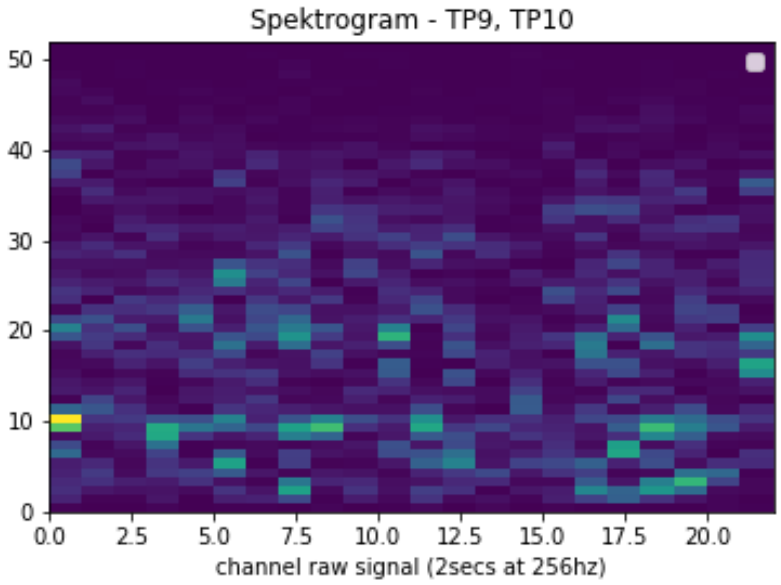
Obr. 2 Výstup po spracovaní 784 png hodnôt

Ako ďalšie sme si pripravili EEG dáta. Na obr. 3 môžeme vidieť vizuálne zobrazené hodnoty týchto dát, konkrétne hodnoty kanálov TP9 a TP10 z jedného merania.



Obr. 3 Vizuálna reprezentácia kanálov TP9 a TP10

Z týchto EEG dát sme vytvorili spektrogramy, ktoré následne posunieme na vstup konvolučnej neurónovej sieti. Spektrogram je 2D vizuálna tepelná mapa, kde vertikálna os predstavuje frekvenčnú os a horizontálna os predstavuje čas signálu. Spektrogram jedného merania môžeme vidieť na nasledujúcom obrázku:



Obr. 4 Spektrogram pre kanály TP9 a TP10

Pripravené modely siete

Model pre obrázkové dáta vo formáte 28x28:

Parametre modelu:

- **loss funkcia:** categorical_crossentropy
- **optimalizátor:** adam
- **počet epoch:** 6
- **batch_size:** 20

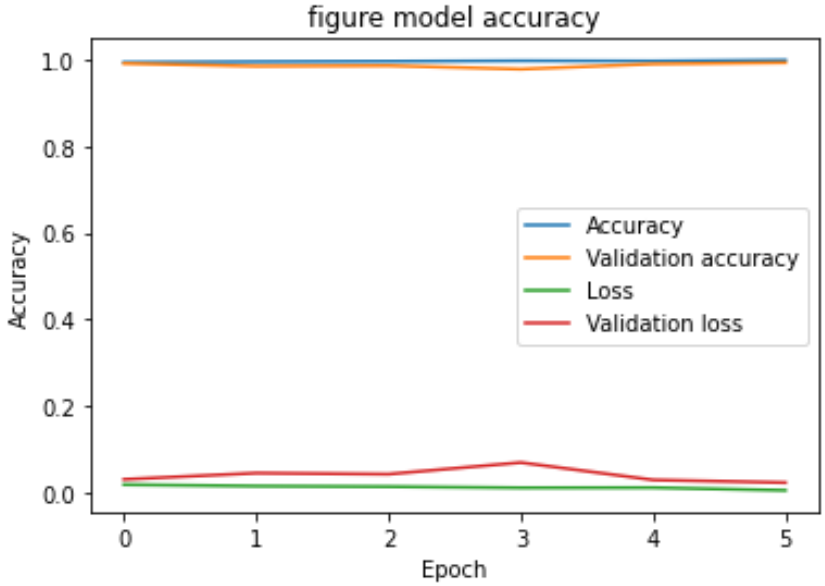
Vrstvy modelu konvolučnej siete:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 24, 24, 32)        832

 max_pooling2d (MaxPooling2D)(None, 12, 12, 32)        0

 dropout (Dropout)           (None, 12, 12, 32)        0

 flatten (Flatten)           (None, 4608)              0

 dense (Dense)               (None, 128)               589952

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 592,074
Trainable params: 592,074
Non-trainable params: 0
```

Priebeh trénovania siete v jednotlivých epochách:

Epoch	Loss	Accuracy
1	0,3159	0,9048
2	0,1047	0,9681
3	0,0614	0,9816
4	0,0414	0,9860
5	0,0366	0,9877
6	0,0220	0,9934

Vidíme, že presnosť siete po poslednej epoche je takmer 100%. Na nasledujúcom grafe je vidieť ako sa menila *presnosť, validačná presnosť, strata* a *validačná strata* v jednotlivých epochách:



Model pre EEG dáta:

Parametre modelu | **konfigurácia 1**:

- **loss funkcia:** categorical_crossentropy
- **optimalizátor:** adam
- **počet epoch:** 10
- **batch_size:** 10

Vrstvy modelu:

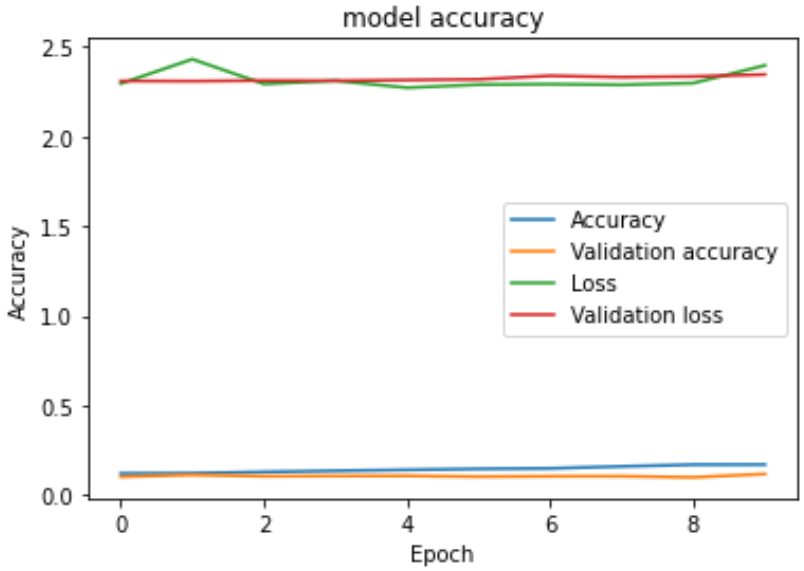
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_1 (Conv2D)            (None, 48, 18, 32)        832

 max_pooling2d_1 (MaxPooling  (None, 24, 9, 32)         0
 2D)

 dropout_1 (Dropout)          (None, 24, 9, 32)         0

 flatten_1 (Flatten)          (None, 6912)              0

 dense_2 (Dense)              (None, 128)               884864

 dense_3 (Dense)              (None, 10)                1290
```

```
=================================================================
Total params: 886,986
Trainable params: 886,986
Non-trainable params: 0
```

Priebeh trénovania siete v jednotlivých epochách:

Epoch	**Loss**	**Accuracy**
1	3,1409	0,1030
2	2,4045	0,1095
3	2,3392	0,1116
4	2,3464	0,1128
5	2,3053	0,1143
6	2,3345	0,1137
7	2,3048	0,1151
8	2,3138	0,1155
9	2,3067	0,1151
10	2,3007	0,1117

Po poslednej epoche je presnosť len na úrovni 11%, čo zobrazuje aj nasledujúci graf:



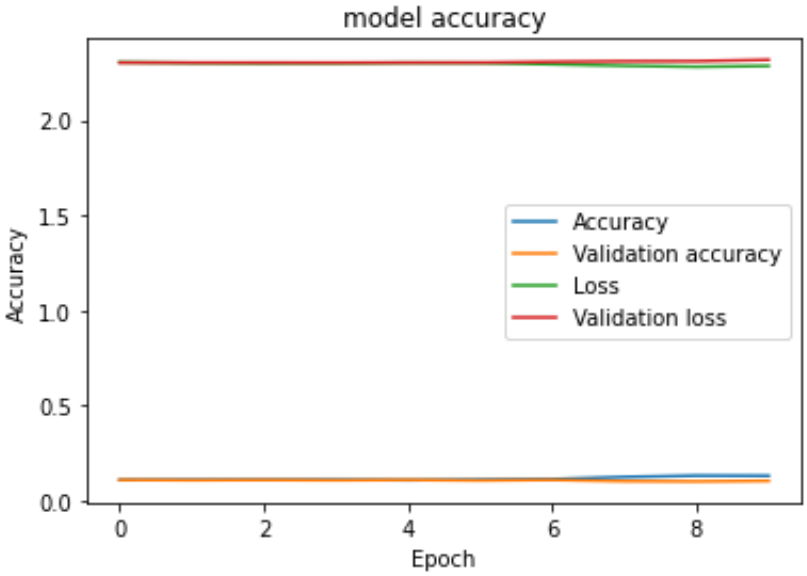
Parametre modelu | **konfigurácia 2**:

Rovnaké parametre ako pre konfiguráciu 1, s rozdielom v konvolučnom jadre, ktoré má veľkosť (3x3) a namiesto `model.add(Dense(128, activation='relu'))` sú štyri vrstvy `model.add(Dense(32, activation='relu'))`

Priebeh trénovania v jednotlivých epochách:

Epoch	**Loss**	**Accuracy**
1	2,4958	0,1014
2	2,3428	0,1098
3	2,3166	0,1107
4	2,3156	0,1096
5	2,3039	0,1108
6	2,3021	0,1111
7	2,3013	0,1115
8	2,3042	0,1113
9	2,3023	0,1107
10	2,3047	0,1104

Vidíme, že konfigurácia 2 dáva takmer rovnaké výsledky ako konfigurácia 1 a to presnosť po desiatich epochách približne 11%, čomu zodpovedá aj nasledujúci graf:



Parametre modelu | **konfigurácia 3 (ResNet50)**:

V tejto konfigurácii sme sa snažili použiť niektorú zo známych sietí, ktoré sú už predtrénované, ako je napríklad sieť ResNet50. Cieľom bolo porovnať výstup takejto natrénovanej siete s výstupmi z našich modelov.

Pri implementácii tohto modelu však nastal problém s tým, že ResNet50 očakáva na vstupe obrázky s tromi kanálmi a navyše šírka a výška obrázka by mala byť aspoň 32. V našom prípade sme ale mali pripravené dáta vo formáte (52, 22, 3), čo nebolo vyhovujúce pre vstup do siete ResNet50. Model teda skončil s nasledujúcou chybou:

```
ValueError: Input size must be at least 32x32; Received: input_shape=(52,
22, 3)
```

Takýto problém nastal aj pri ďalších predtrénovaných sieťach, ako je napríklad VGG16, VGG19 či MobileNet, ktoré sme tiež vyskúšali.

Pokúšali sme sa túto chybu odstrániť pridaním *paddingu*/*výplne* okolo obrázka, aby obrázok získal potrebný formát, čo však skončilo iba ďalšou chybou a teda neúspešne.

Implementácia modelu v Python-e vyzerá nasledovne:

```
def resnet_model():
  res_model = Sequential()

  pretrained_model = tf.keras.applications.ResNet50(
      include_top=False,
      weights='imagenet',
      pooling='max',
      input_shape=(52, 22, 3)
  )


  for layer in pretrained_model.layers:
    layer.trainable = False

  res_model.add(pretrained_model)
  res_model.add(Flatten())
  res_model.add(Dense(128, activation='relu'))
  res_model.add(Dense(10, activation='softmax'))

  res_model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])

  return res_model
```

Zhrnutie

Ako bolo vidieť pri rôznych konfiguráciách, trénovanie siete na EEG dátach neposkytlo dostatočnú presnosť. Úspešnosť približne 11%, ktorú sa podarilo dosiahnuť, môžeme považovať za zlú. Tieto dáta sa preto javia ako nie veľmi vhodné pre klasifikáciu cifier.

Trénovanie siete na dátach, ktoré reprezentujú obrázok vo formáte 28x28 (784 hodnôt) dopadlo s úspešnosťou takmer 100%.