

Temă Învățare Automată

Raport 2

Setup

Creare Tensors

```
[17] def gestures_to_tensor(x):  
    nr, nc = x.shape  
    arr = []  
    for i in range(nr):  
        row = x.iloc[i]  
        arr.append(list(zip(row[0], row[1], row[2])))  
    return tf.convert_to_tensor(arr)  
  
[18] tf_gst_train_x = gestures_to_tensor(gestures_train_x)  
tf_gst_train_y = to_categorical(list(map(lambda str : float(str)-1, gestures_train_y)))  
tf_gst_test_x = gestures_to_tensor(gestures_test_x)  
tf_gst_test_y = to_categorical(list(map(lambda str : float(str)-1, gestures_test_y)))
```

LSTM

Definirea modelului

```
def gst_define_lstm_model(type='lcell', optimizer='adam'):  
    n_examples = gestures_train_x.shape[0]  
    n_features = gestures_train_x.shape[1]  
    n_timesteps = len(gestures_train_x.iloc[0, 0])  
  
    lstm_model = Sequential()  
  
    if type == 'lcell':  
        lstm_model.add(LSTM(150, input_shape=(n_timesteps, n_features)))  
    elif type == '2cells':  
        lstm_model.add(LSTM(100, input_shape=(n_timesteps, n_features), return_sequences=True))  
        lstm_model.add(LSTM(50))  
    elif type == 'bidir':  
        lstm_model.add(Bidirectional(LSTM(150, input_shape=(n_timesteps, n_features))))  
  
    lstm_model.add(Dropout(0.2))  
    lstm_model.add(Dense(100, activation='relu'))  
    lstm_model.add(Dense(8, activation='softmax'))  
    lstm_model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])  
  
    return lstm_model
```

Var. 1 LSTM

Arhitectură Model

- Straturi:
 - Un strat LSTM cu 150 unități ascunse;
 - Un strat Dropout cu procentul setat la 0.2;
 - ReLU;
 - Softmax pentru a calcula clasa.
- Validarea se face prin funcția de Cross Entropy.
- Optimizatorul ADAM.
- 50 epoci cu batch_size 128.

Rezultate pe datele de test

	precision	recall	f1-score	support
1	0.85	0.74	0.79	437
2	0.87	0.94	0.90	452
3	0.92	0.85	0.88	454
4	0.86	0.64	0.73	450
5	0.72	0.97	0.83	433
6	0.63	0.54	0.58	449
7	0.69	0.75	0.72	447
8	0.85	0.95	0.90	460
accuracy			0.80	3582
macro avg	0.80	0.80	0.79	3582
weighted avg	0.80	0.80	0.79	3582

Matricea de confuzie

	pred:1	pred:2	pred:3	pred:4	pred:5	pred:6	pred:7	pred:8
true:1	323	35	0	2	0	53	24	0
true:2	16	423	0	0	0	3	8	2
true:3	0	0	385	3	31	7	1	27
true:4	4	0	4	287	79	16	52	8
true:5	0	2	5	2	422	2	0	0
true:6	15	9	20	17	47	242	63	36
true:7	18	17	2	13	2	56	336	3
true:8	2	0	3	8	2	8	1	436

Var. 2 LSTM

Arhitectură Model

- Straturi:
 - Un strat LSTM cu 100 unități ascunse;
 - Un strat LSTM cu 50 unități ascunse;
 - Un strat Dropout cu procentul setat la 0.2;
 - ReLU;
 - Softmax pentru a calcula clasa.
- Validarea se face prin funcția de Cross Entropy.
- Optimizatorul ADAM.
- 50 epoci cu batch_size 128.

Rezultate pe datele de test

	precision	recall	f1-score	support
1	0.84	0.75	0.79	437
2	0.93	0.96	0.95	452
3	0.86	0.97	0.91	454
4	0.85	0.81	0.83	450
5	0.79	0.96	0.87	433
6	0.67	0.54	0.60	449
7	0.84	0.84	0.84	447
8	0.94	0.91	0.92	460
accuracy			0.84	3582
macro avg	0.84	0.84	0.84	3582
weighted avg	0.84	0.84	0.84	3582

Matricea de confuzie

	pred:1	pred:2	pred:3	pred:4	pred:5	pred:6	pred:7	pred:8
true:1	323	35	0	2	0	53	24	0
true:2	16	423	0	0	0	3	8	2
true:3	0	0	385	3	31	7	1	27
true:4	4	0	4	287	79	16	52	8
true:5	0	2	5	2	422	2	0	0
true:6	15	9	20	17	47	242	63	36
true:7	18	17	2	13	2	56	336	3
true:8	2	0	3	8	2	8	1	436

Var. 3 LSTM

Arhitectură Model

- Straturi:
 - Un strat bidirecțional LSTM cu 150 unități ascunse;
 - Un strat Dropout cu procentul setat la 0.2;
 - ReLU;
 - Softmax pentru a calcula clasa.
- Validarea se face prin funcția de Cross Entropy.
- Optimizatorul ADAM.
- 50 epoci cu batch_size 128.

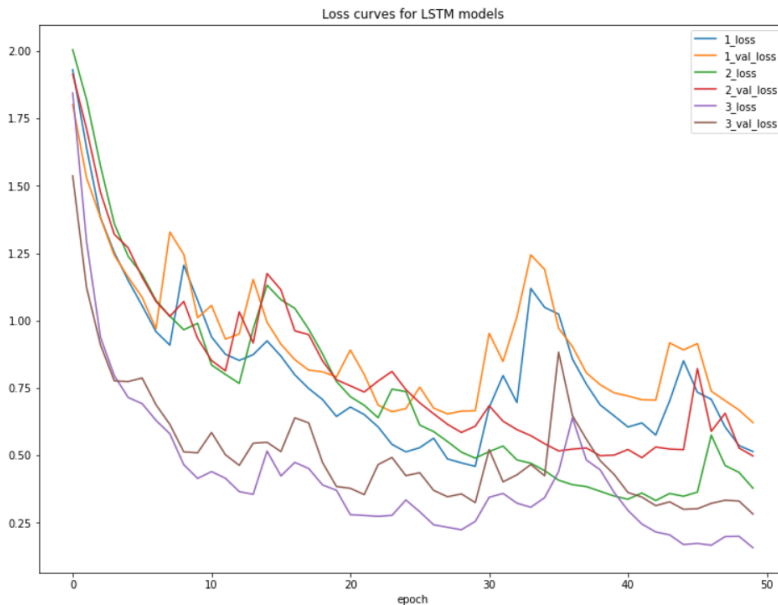
Rezultate pe datele de test

	precision	recall	f1-score	support
1	0.88	0.89	0.89	437
2	0.99	0.97	0.98	452
3	0.91	0.93	0.92	454
4	0.89	0.87	0.88	450
5	0.86	0.94	0.90	433
6	0.86	0.77	0.82	449
7	0.92	0.95	0.93	447
8	0.97	0.96	0.96	460
accuracy			0.91	3582
macro avg	0.91	0.91	0.91	3582
weighted avg	0.91	0.91	0.91	3582

Matricea de confuzie

	pred:1	pred:2	pred:3	pred:4	pred:5	pred:6	pred:7	pred:8
true:1	391	0	0	2	0	27	17	0
true:2	3	437	0	0	0	0	10	2
true:3	3	0	424	0	15	10	2	0
true:4	9	1	3	390	23	15	5	4
true:5	3	0	9	11	406	3	1	0
true:6	34	0	23	16	28	347	0	1
true:7	2	4	4	7	1	0	423	6
true:8	0	0	1	13	1	0	4	441

Curbele de loss pentru cele 3 variante de model LSTM



Concluzii LSTM

- Multe confundări între clase, cu clasele 4, 6 și 7 fiind cele mai grele de clasificat corect.
- Unele clase au recall foarte bun, dar precizia mai mică (ex. clasa 5).
- Folosirea a 2 layers LSTM duce la o îmbunătățire a acurateții.
- Modelul cu celulă LSTM bidirecțională are cele mai bune rezultate.
- Din curbele de loss se observă că modelele nu fac overfit și reușesc să clasifice și date noi.
- În curbe apar "spikes" datorită folosirii de mini-batches.

MLP

Var. 1 MLP

Arhitectură Model

- Retea Multi-Layered Perceptron cu straturile:
 - Un start de liniarizare.
 - 2 straturi dense de dimensiuni 256 și 128, folosind funcția de activare **sigmoid**.
 - 1 start dens cu funcția **softmax**.
- Validarea se face prin funcția de Cross Entropy.
- Optimizatorul **ADAM**.
- 100 epoci cu batch_size 128.

Rezultate pe datele de test

	precision	recall	f1-score	support
1	0.98	0.94	0.96	437
2	1.00	0.97	0.98	452
3	0.96	0.95	0.96	454
4	0.95	0.90	0.92	450
5	0.86	0.97	0.91	433
6	0.88	0.88	0.88	449
7	0.96	0.97	0.97	447
8	0.97	0.98	0.98	460
accuracy			0.94	3582
macro avg	0.95	0.94	0.94	3582
weighted avg	0.95	0.94	0.94	3582

Matricea de confuzie

	pred:1	pred:2	pred:3	pred:4	pred:5	pred:6	pred:7	pred:8
true:1	409	0	0	1	0	20	6	1
true:2	1	439	0	0	0	1	9	2
true:3	0	0	431	0	10	10	3	0
true:4	0	0	1	403	24	17	0	5
true:5	2	0	6	1	420	4	0	0
true:6	4	0	6	12	30	394	0	3
true:7	3	1	2	1	0	4	435	1
true:8	0	0	1	4	2	0	0	453

Var. 2 MLP

Arhitectură Model

- Retea Multi-Layered Perceptron cu straturile:
 - Un start de liniarizare.
 - 2 straturi dense de dimensiuni 256 si 128, folosind funcția de activare **ReLU**.
 - 1 start dens cu funcția **softmax**.
- Validarea se face prin funcția de Cross Entropy.
- Optimizatorul **Stochastic Gradient Descent**.
- 100 epoci cu batch_size 128.

Rezultate pe datele de test

	precision	recall	f1-score	support
1	0.97	0.95	0.96	437
2	0.98	0.98	0.98	452
3	0.97	0.93	0.95	454
4	0.96	0.86	0.91	450
5	0.85	0.96	0.90	433
6	0.87	0.89	0.88	449
7	0.98	0.98	0.98	447
8	0.98	0.99	0.99	460
accuracy			0.94	3582
macro avg	0.95	0.94	0.94	3582
weighted avg	0.95	0.94	0.94	3582

Matricea de confuzie

	pred:1	pred:2	pred:3	pred:4	pred:5	pred:6	pred:7	pred:8
true:1	416	2	0	0	0	19	0	0
true:2	1	445	0	1	0	1	3	1
true:3	1	2	423	0	9	12	6	1
true:4	2	0	1	387	36	22	0	2
true:5	3	1	7	1	416	4	0	1
true:6	5	1	4	10	26	400	1	2
true:7	1	1	3	1	0	2	439	0
true:8	0	0	0	4	1	0	0	455

Var. 3 MLP

Arhitectură Model

- Retea Multi-Layered Perceptron cu straturile:
 - Un start de liniarizare.
 - 3 straturi dense de dimensiuni 512, 256 si 128, folosind funcția de activare **ReLU**, fiecare urmat de un start de Dropout cu procentul 0.1.
 - 1 start dens cu funcția **softmax**.
- Validarea se face prin funcția de Cross Entropy.
- Optimizatorul **ADAM**.
- 100 epoci cu batch_size 128.

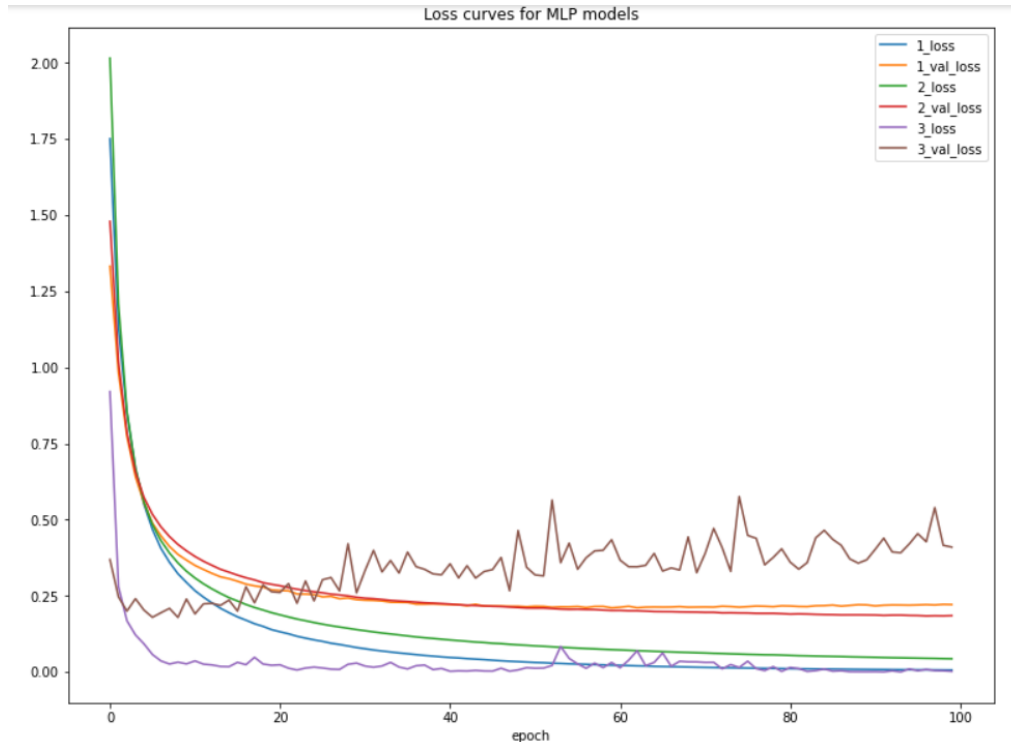
Rezultate pe datele de test

	precision	recall	f1-score	support
1	0.97	0.95	0.96	437
2	1.00	0.99	0.99	452
3	0.97	0.94	0.95	454
4	0.94	0.90	0.92	450
5	0.89	0.97	0.93	433
6	0.88	0.90	0.89	449
7	0.99	1.00	0.99	447
8	0.99	0.99	0.99	460
accuracy			0.95	3582
macro avg	0.95	0.95	0.95	3582
weighted avg	0.95	0.95	0.95	3582

Matricea de confuzie

	pred:1	pred:2	pred:3	pred:4	pred:5	pred:6	pred:7	pred:8
true:1	409	0	0	1	0	20	6	1
true:2	1	439	0	0	0	1	9	2
true:3	0	0	431	0	10	10	3	0
true:4	0	0	1	403	24	17	0	5
true:5	2	0	6	1	420	4	0	0
true:6	4	0	6	12	30	394	0	3
true:7	3	1	2	1	0	4	435	1
true:8	0	0	1	4	2	0	0	453

Curbele de loss pentru cele 3 variante de model MLP



Concluzii MLP

- Rezultate foarte bune printr-o rețea simplă complet-conectată.
- Rezultate ușor mai bune pentru optimizarea cu Adam față de SGD.
- Datele generalizează bine, iar adăugarea de straturi de Dropout nu este necesară și duce chiar la niște pierderi.
- Gestul 6 este cel mai greu de clasificat corect, probabil din cauza simplității sale și asemănării cu alte gesturi, cum ar fi 5 și 1.

Rețele convoluționale

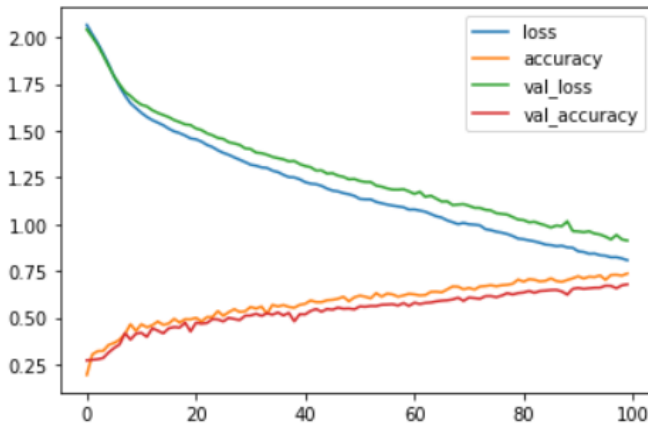
Var. 1 CNN

Arhitectură Model

- Retea cu straturile:
 - Un start de reshape, output_shape=(315, 3).
 - Un start convoluțional, cu 256 de filtre de dimensiune 5 și padding, folosind funcția de activare **ReLU**, output_shape=(315, 256).
 - Un start de Global Average Pooling, output_shape=(256).

- Un strat dens de dimensiune 100 cu funcția **ReLU**.
- 1 strat dens cu funcția **softmax**.
- Validarea se face prin funcția de Cross Entropy.
- Optimizatorul **ADAM**.
- 100 epoci cu batch_size 128.

Rezultate antrenare



- Acuratețea crește, dar mult prea încet (se ajunge doar la un 0.6-0.7 după 100 de epoci).
Vom experimente cu mai multe straturi liniare finale și cu straturi convoluționale de dimensiuni diferite. Putem eventual crește numărul de epoci.

Var. 2 CNN

Arhitectură Model

- Retea cu straturile:
 - Un strat de reshape, output_shape=(315, 3).
 - Un strat convoluțional, cu 256 de filtre de dimensiune 5 și padding, folosind funcția de activare **ReLU**, output_shape=(315, 256).
 - Un strat de Global Average Pooling, output_shape=(256).
 - Un strat dens de dimensiune 256 cu funcția **ReLU**.
 - Un strat dens de dimensiune 128 cu funcția **ReLU**.
 - 1 strat dens cu funcția **softmax**.
- Validarea se face prin funcția de Cross Entropy.
- Optimizatorul **ADAM**.
- 100 epoci cu batch_size 128.

Rezultate pe datele de test

	precision	recall	f1-score	support
1	0.61	0.74	0.67	437
2	0.83	0.85	0.84	452
3	0.84	0.33	0.47	454
4	0.56	0.77	0.65	450
5	0.58	0.61	0.59	433
6	0.64	0.53	0.58	449
7	0.77	0.91	0.84	447
8	0.92	0.88	0.90	460
accuracy			0.70	3582
macro avg	0.72	0.70	0.69	3582
weighted avg	0.72	0.70	0.69	3582

Matricea de confuzie

	pred:1	pred:2	pred:3	pred:4	pred:5	pred:6	pred:7	pred:8
true:1	325	26	2	27	24	3	28	2
true:2	40	383	0	0	0	2	25	2
true:3	52	6	148	187	15	8	12	26
true:4	38	2	10	347	19	8	23	3
true:5	23	8	4	7	263	111	17	0
true:6	33	3	7	25	124	240	15	2
true:7	8	8	0	19	0	4	408	0
true:8	10	28	5	3	9	0	1	404

Var. 3 CNN

Arhitectură Model

- Retea cu straturile:
 - Un start de reshape, **care creează ferestre de lungime 35 pentru seriile de timp, output_shape=(35, 3*(315/35))**.
 - Un start convoluțional, cu 64 de filtre de dimensiune 5 și padding, folosind funcția de activare **ReLU**, output_shape=(35, 64).
 - Un start convoluțional, cu 32 de filtre de dimensiune 3, folosind funcția de activare **ReLU**, output_shape=(33, 32).
 - Un start de Global Average Pooling, output_shape=(32).
 - Un strat dens de dimensiune 256 cu funcția **sigmoid**.
 - Un strat dens de dimensiune 128 cu funcția **sigmoid**.
 - 1 strat dens cu funcția **softmax**.
- Validarea se face prin funcția de Cross Entropy.
- Optimizatorul **ADAM**.

- 100 epoci cu batch_size 128.

Rezultate pe datele de test

	precision	recall	f1-score	support
1	0.96	0.98	0.97	437
2	1.00	1.00	1.00	452
3	0.98	0.93	0.95	454
4	0.95	0.93	0.94	450
5	0.90	0.98	0.94	433
6	0.96	0.91	0.93	449
7	0.96	1.00	0.98	447
8	1.00	0.98	0.99	460
accuracy			0.96	3582
macro avg	0.96	0.96	0.96	3582
weighted avg	0.96	0.96	0.96	3582

Matricea de confuzie

	pred:1	pred:2	pred:3	pred:4	pred:5	pred:6	pred:7	pred:8
true:1	427	0	1	5	0	4	0	0
true:2	1	451	0	0	0	0	0	0
true:3	2	0	421	7	6	10	6	2
true:4	9	0	0	418	18	3	2	0
true:5	1	0	4	2	424	1	1	0
true:6	4	0	3	4	21	409	8	0
true:7	0	0	1	1	0	0	445	0
true:8	3	0	0	4	0	1	1	451

Var. 4 CNN

Arhitectură Model

- Retea cu straturile:
 - Un start de reshape, **care creează ferestre de lungime 105 pentru seriile de timp, output_shape=(105, 3*(315/105))**.
 - Un start convoluțional, cu 128 de filtre de dimensiune 5 și padding, folosind funcția de activare **ReLU**, output_shape=(105, 128).
 - Un start convoluțional, cu 64 de filtre de dimensiune 3, folosind funcția de activare **ReLU**, output_shape=(103, 64).
 - Un start de Global Average Pooling, output_shape=(64).
 - Un strat dens de dimensiune 64 cu funcția **ReLU**.
 - Un strat dens de dimensiune 32 cu funcția **ReLU**.

- 1 start dens cu funcția **softmax**.
- Validarea se face prin funcția de Cross Entropy.
- Optimizatorul **ADAM**.
- 100 epoci cu batch_size 128.

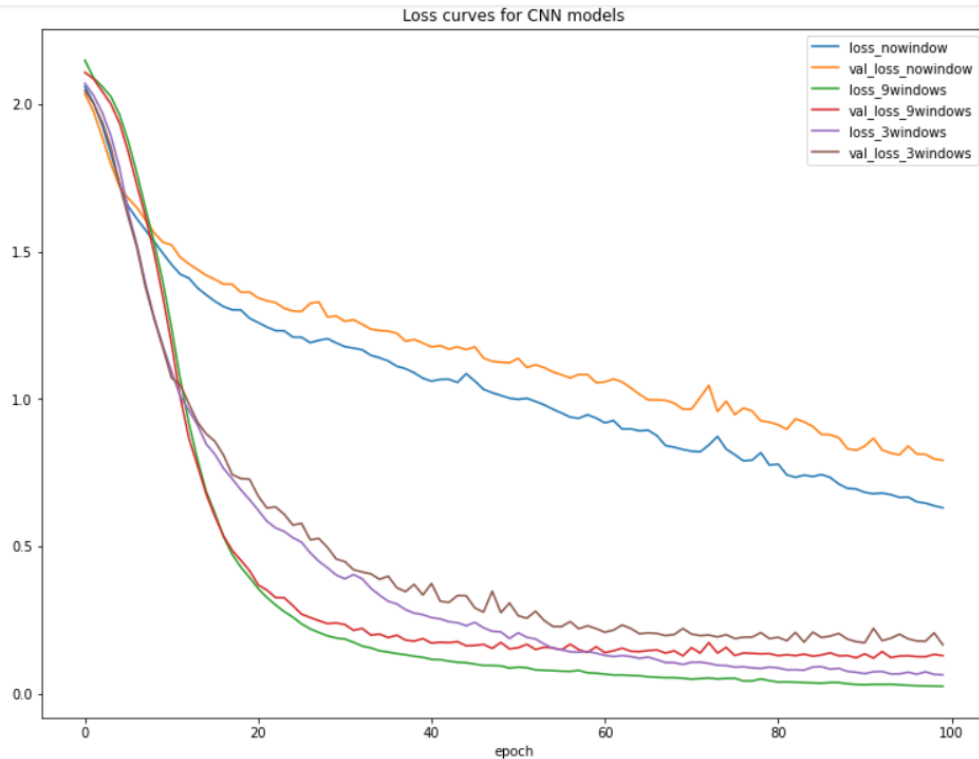
Rezultate pe datele de test

	precision	recall	f1-score	support
1	0.92	0.95	0.94	437
2	0.98	0.98	0.98	452
3	0.92	0.95	0.93	454
4	0.93	0.94	0.93	450
5	0.95	0.97	0.96	433
6	0.97	0.90	0.93	449
7	0.96	0.95	0.95	447
8	0.99	0.98	0.99	460
accuracy			0.95	3582
macro avg	0.95	0.95	0.95	3582
weighted avg	0.95	0.95	0.95	3582

Matricea de confuzie

	pred:1	pred:2	pred:3	pred:4	pred:5	pred:6	pred:7	pred:8
true:1	332	21	9	29	1	8	21	16
true:2	14	415	4	6	8	2	3	0
true:3	31	8	245	107	11	19	12	21
true:4	28	5	114	244	22	7	4	26
true:5	5	12	6	8	296	89	16	1
true:6	17	10	14	21	125	239	13	10
true:7	18	16	24	5	4	3	288	89
true:8	11	6	8	25	3	5	52	350

Curbele de loss pentru Var. 2, 3 și 4 CNN



Concluzii CNN

- Aplicarea filtrelor pe întreaga secvență de date duce la rezultate mai proaste, pentru că se pierde prea multe informații atunci când aplică stratul de pooling. Doar 3 canale inițiale nu sunt suficiente pentru rezultate OK în 100 epoci. Totuși modelul încă se îmbunătățește treptat, deci antrenarea pe mai multe epoci ar fi dus probabil la rezultate mai OK.
- Folosirea de ferestre, considerate canale separate de date, duce la rezultate foarte bine. Complexitatea intrărilor crește și nu se mai pierde la fel de multă informație în stratul de pooling. Se observă și că nu este o diferență mare între 3 sau 9 ferestre.

Rezultate finale

	General				1			2			3			4			5			6			7			8		
Arhitectura	A	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LSTM [LSTM(150), Dropout(0.2), ReLU(100)] Adam 50 epoci, B=128	0.8	0.8	0.8	0.79	0.85	0.74	0.79	0.87	0.94	0.9	0.92	0.85	0.88	0.86	0.64	0.73	0.72	0.97	0.83	0.63	0.54	0.58	0.69	0.75	0.72	0.85	0.95	0.90
LSTM [LSTM(100), LSTM(50), Dropout(0.2), ReLU(100)] ADAM 50 epoci, B=128	0.84	0.84	0.84	0.84	0.84	0.75	0.79	0.93	0.96	0.95	0.86	0.97	0.91	0.85	0.81	0.83	0.79	0.96	0.87	0.67	0.54	0.6	0.84	0.84	0.84	0.94	0.91	0.92
LSTM [BLSTM(150)], Dropout(0.2), ReLU(100)] ADAM 50 epoci, B=128	0.91	0.91	0.91	0.91	0.88	0.89	0.89	0.99	0.97	0.98	0.91	0.93	0.92	0.89	0.87	0.88	0.86	0.94	0.9	0.86	0.77	0.82	0.92	0.95	0.93	0.97	0.96	0.96
MLP [Flatten, Sigmoid(256), Sigmoid(128)] ADAM 100 epoci, B=128	0.94	0.94	0.94	0.94	0.98	0.94	0.96	1	0.97	0.98	0.96	0.95	0.96	0.95	0.9	0.92	0.86	0.97	0.91	0.88	0.88	0.88	0.96	0.97	0.97	0.97	0.98	0.98
MLP [Flatten, ReLU(256), ReLU(128)] SGD 100 epoci, B=128	0.94	0.94	0.94	0.94	0.97	0.95	0.96	0.98	0.98	0.98	0.97	0.93	0.95	0.96	0.86	0.91	0.85	0.96	0.9	0.87	0.89	0.88	0.98	0.98	0.98	0.98	0.99	0.99
MLP [Flatten, ReLU(512), ReLU(256), ReLU(128), Dropout(0.1)] ADAM 100 epoci, B=128	0.95	0.95	0.95	0.95	0.97	0.95	0.96	1	0.99	0.99	0.97	0.94	0.95	0.94	0.9	0.92	0.89	0.97	0.93	0.88	0.9	0.89	0.99	1	0.99	0.99	0.99	0.99
CNN [Conv1D(256, 5), GlobalAvgPool, ReLU(256), ReLU(128)] ADAM 100 epoci, B=128	0.7	0.72	0.7	0.69	0.61	0.74	0.67	0.83	0.85	0.84	0.84	0.33	0.47	0.56	0.77	0.65	0.58	0.61	0.59	0.64	0.53	0.58	0.77	0.91	0.84	0.92	0.88	0.9
CNN, 9 ferestre [Conv(64, 5), Conv(32, 3), GlobalAvgPool, ReLU(256), ReLU(128)]	0.96	0.96	0.96	0.96	0.96	0.98	0.97	1	1	1	0.98	0.93	0.95	0.95	0.93	0.94	0.9	0.98	0.94	0.96	0.91	0.93	0.96	1	0.98	1	0.98	0.99
CNN, 3 ferestre [Conv(128, 5), Conv(64, 3), GlobalAvgPool, ReLU(64), ReLU(32)]	0.95	0.95	0.95	0.95	0.92	0.95	0.94	0.98	0.98	0.98	0.92	0.95	0.93	0.93	0.94	0.93	0.95	0.97	0.96	0.97	0.9	0.93	0.96	0.95	0.95	0.99	0.98	0.99

- Arhitectura simplă, de tip MLP, are rezultate foarte bune. Prin straturi liniari complet-conectate, modelul învață suficienți parametrii pentru a reuși să clasifice corect.
- Atunci când am crescut complexitatea (nr. de parametrii), am adăugat nivele de Dropout pentru a nu ajunge la overfitting.

- Rețeaua recurentă cu layer de tip LSTM are performanțe scăzute, deci deși este un model mai "sophisticat", nu este neapărat mai bun pentru problemă. Plus că este mult mai încet. De asemenea, curbele de loss indică faptul că mai multe epoci nu ar fi dus neapărat la rezultate mai bune. Este și foarte influențat de mini-batches proaste, în curbe apărând spikes des.
- Rețeaua convoluțională are cele mai proaste rezultate atunci când filtrele se aplică asupra întregii secvențe pentru că se pierde mult prea multe informații prin stratul de pooling. Totuși, curbele de loss arată că rețeaua se îmbunătățește treptat și stabil, deci prin mai multe epoci s-ar putea ajunge la un model care să clasifice corect.
- Modele CNN în care am separat secvențele în mai multe ferestre, considerate canale diferite, și am folosit mai multe straturi mari liniari au rezultatele cele mai bune.
- Unele gesturi sunt mai greu de clasificat, cum ar fi 6. Altele se disting mai mult, cum ar fi 2 și 8.