

Proiect Inteligență Artificială

- documentație -

Pentru rezolvarea cerinței acestui proiect am dezvoltat 3 modele de inteligență artificială cu ajutorul cărora am reușit să obțin diferite scoruri de acuratețe pentru datele de validare:

1. Un model Multinomial Naive-Bayes dezvoltat folosind biblioteca `scikit-learn.naive_bayes`
2. Un model bazat pe o rețea de percepștroni dezvoltată folosind `MLPClassifier` din biblioteca `scikit-learn`
3. Un model bazat pe o rețea neuronală convolușională dezvoltată folosind biblioteca `Tensorflow`

Pentru citirea datelor (atât cele de antrenare, cât și cele de validare și testare) am folosit clasa `Image` din biblioteca `PIL` și metoda `open` a acestei clase. Pentru toate cele 3 modele am nevoie de matrici bidimensionale având mărimea de $\text{nr_imagini} * \text{dimensiune_imagini}$. Am observant că dimensiunea imaginilor este de $32 * 32 = 1024$, iar numărul de imagini din fiecare set de date putea fi numărat cu ajutorul fișierelor de tip `txt` (`train.txt`, `validation.txt`, `test.txt`). Astfel, am reușit să creez 3 matrici cu dimensiunile 30001×1024 (pentru `train`), 5000×1024 (pentru `validation`), respectiv 5000×1024 (pentru `test`).

Mai departe voi descrie pașii efectuați pentru dezvoltarea fiecărui model încercat:

1. Multinomial Naive-Bayes

Înainte de toate, am observant că valorile pixelilor din mulțimile date (`train`, `validation`, `test`) sunt cuprinse între 0 și 255 (inclusiv) și am definit o funcție de discretizare a mulțimilor noastre care primește ca parametru, pe lângă matricea cu datele imaginilor, și intervalele împărțite între 0 și 255, obținute folosind funcția `linspace` din `Numpy`. Funcția a fost definită cu ajutorul funcției `digitize` din `Numpy`. După definire, am apelat funcția cu valori cuprinse între 1 și 15 pentru numărul de intervale folosite. Pentru fiecare valoare returnată de funcția de discretizare am creat un model folosind clasa `MultinomialNB` din `sklearn.naive_bayes`, am antrenat modelul cu metodă `fit` a clasei, iar cu ajutorul metodelor `predict` și `score` ale aceleiași clase am obținut etichetele prezise de model pentru mulțimea de validare, respectiv acuratețea cu care aceste etichete au fost prezise. Cu ajutorul mulțimilor de etichete am reprezentat matricea de confuzie pentru fiecare model în parte.

Rezultatele sunt urmatoarele:

- Nr. Intervale = 1:

Acuratețe = 0.1108

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	0	0	0	0	570	0	0	0	0
1:	0	0	0	0	527	0	0	0	0
2:	0	0	0	0	533	0	0	0	0
3:	0	0	0	0	578	0	0	0	0
4:	0	0	0	0	554	0	0	0	0
5:	0	0	0	0	561	0	0	0	0
6:	0	0	0	0	580	0	0	0	0
7:	0	0	0	0	520	0	0	0	0
8:	0	0	0	0	577	0	0	0	0

- Nr. Intervale = 2:

Acuratețe = 0.157

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	7	45	8	16	462	13	8	1	10
1:	8	106	21	29	304	16	17	7	19
2:	1	27	27	23	383	48	11	2	11
3:	0	37	14	31	408	60	5	2	21
4:	7	25	11	29	424	28	15	0	15
5:	0	15	21	17	395	66	18	3	26
6:	10	60	6	14	353	39	65	2	31
7:	3	67	19	20	351	21	19	8	12
8:	0	61	4	15	372	41	29	4	51

- Nr. Intervale = 3:

Acuratețe = 0.3582

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	58	84	20	46	103	59	36	80	84
1:	20	218	26	51	24	27	27	87	47
2:	16	51	139	38	50	120	28	66	25
3:	19	50	9	276	40	98	15	46	25
4:	17	47	43	44	178	90	18	88	29
5:	11	12	27	54	40	348	8	42	19
6:	34	61	15	52	109	45	134	55	75
7:	19	68	24	36	68	33	26	226	20
8:	44	47	17	52	65	81	30	27	214

- Nr. Intervale = 4:

Acuratețe = 0.3754

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	53	85	24	47	96	41	41	94	89
1:	15	226	24	50	18	22	28	99	45
2:	21	63	147	32	42	110	23	70	25
3:	17	56	10	289	40	80	19	45	22
4:	23	44	41	40	185	82	19	98	22
5:	8	19	29	59	35	342	10	41	18
6:	42	62	16	49	90	35	143	61	82
7:	23	57	22	35	62	25	23	256	17
8:	45	47	15	55	56	62	32	29	236

- Nr. Intervale = 5:

Acuratețe = 0.3856

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	55	86	26	43	96	34	41	97	92
1:	13	241	23	46	17	21	25	97	44
2:	18	62	153	28	43	104	29	75	21
3:	19	57	8	297	38	73	24	41	21
4:	25	48	41	36	182	77	21	96	28
5:	6	17	30	55	36	342	10	48	17
6:	37	67	18	47	91	33	143	65	79
7:	21	53	21	37	60	26	19	267	16
8:	44	56	15	51	51	54	26	32	248

- Nr. Intervale = 6:

Acuratețe = 0.3878

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	60	87	23	42	87	33	41	102	95
1:	15	244	22	44	15	23	23	97	44
2:	20	63	147	27	45	98	32	79	22
3:	19	57	8	298	37	69	22	45	23
4:	21	48	42	39	179	75	20	103	27
5:	7	16	34	52	39	340	10	45	18
6:	36	66	13	48	92	30	152	66	77
7:	19	56	20	34	62	22	19	271	17
8:	46	50	16	49	56	52	28	32	248

- Nr. Intervale = 7:

Acuratețe = 0.3894

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	60	86	26	44	91	34	42	96	91
1:	13	246	24	42	15	20	25	97	45
2:	18	61	151	29	45	97	31	77	24
3:	20	55	8	303	35	66	23	44	24
4:	21	47	40	41	180	72	19	106	28
5:	6	17	29	59	38	339	11	45	17
6:	39	67	16	49	87	30	144	69	79
7:	17	53	23	33	62	23	21	272	16
8:	45	50	16	49	54	51	27	33	252

- Nr. Intervale = 8:

Acuratețe = 0.3874

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	60	89	24	45	90	32	43	95	92
1:	16	242	22	43	15	20	22	101	46
2:	19	61	147	29	42	102	31	80	22
3:	20	57	8	299	35	68	24	44	23
4:	22	49	40	40	177	74	19	105	28
5:	6	16	28	56	37	340	11	51	16
6:	41	63	12	48	89	29	148	73	77
7:	19	57	18	32	61	24	19	274	16
8:	43	52	17	51	53	46	31	34	250

- Nr. Intervale = 9:

Acuratețe = 0.391

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	60	88	23	45	89	30	41	101	93
1:	13	240	25	43	14	20	24	104	44
2:	20	65	146	25	46	100	31	78	22
3:	18	57	9	301	35	67	23	45	23
4:	21	49	38	42	182	71	18	105	28
5:	5	16	31	55	36	338	13	52	15
6:	40	64	11	50	88	30	151	71	75
7:	20	51	20	34	56	21	19	285	14
8:	45	50	17	51	51	46	31	34	252

- Nr. Intervale = 10:

Acuratețe = 0.3888

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	56	89	26	45	92	30	41	99	92
1:	14	243	23	43	15	21	23	100	45
2:	20	65	149	26	45	99	30	78	21
3:	19	57	9	298	33	68	24	46	24
4:	21	48	41	40	183	69	20	104	28
5:	6	15	32	56	37	337	12	50	16
6:	42	65	11	50	89	29	146	70	78
7:	19	54	18	33	60	21	18	282	15
8:	47	53	15	52	50	44	32	34	250

- Nr. Intervale = 11:

Acuratețe = 0.3916

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	63	87	22	46	89	28	43	99	93
1:	14	242	23	43	15	21	24	100	45
2:	21	64	148	26	44	98	31	79	22
3:	19	58	8	298	34	68	23	46	24
4:	20	49	40	38	180	73	20	106	28
5:	6	15	31	54	37	337	13	52	16
6:	42	64	11	48	86	29	155	70	75
7:	20	53	21	33	56	22	17	283	15
8:	48	54	16	48	51	44	29	35	252

- Nr. Intervale = 12:

Acuratețe = 0.392

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	63	88	25	45	87	28	43	100	91
1:	16	239	24	43	14	21	23	103	44
2:	19	64	148	28	43	99	32	79	21
3:	18	60	8	298	34	68	23	46	23
4:	19	50	40	39	183	71	20	104	28
5:	6	15	31	56	37	335	13	52	16
6:	43	64	11	48	89	29	148	72	76
7:	18	53	19	33	52	22	18	290	15
8:	46	52	16	48	52	46	30	31	256

- Nr. Intervale = 13:

Acuratețe = 0.3906

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	60	86	25	46	87	28	43	102	93
1:	16	242	23	42	15	21	23	101	44
2:	20	64	150	26	45	96	31	78	23
3:	19	58	9	297	34	68	22	46	25
4:	19	49	41	39	182	70	20	106	28
5:	6	15	31	59	37	332	13	51	17
6:	40	67	10	47	93	29	146	71	77
7:	18	54	20	33	55	22	19	285	14
8:	47	50	16	47	51	43	30	34	259

- Nr. Intervale = 14:

Acuratețe = 0.3908

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	63	87	24	45	89	30	41	99	92
1:	16	240	23	42	15	21	24	102	44
2:	18	68	147	25	45	98	31	80	21
3:	18	58	9	300	33	66	22	47	25
4:	18	51	41	38	182	72	20	104	28
5:	6	15	31	56	37	334	13	52	17
6:	46	66	11	47	87	29	147	71	76
7:	20	54	18	34	55	21	18	286	14
8:	50	50	15	48	51	43	32	33	255

- Nr. Intervale = 15:

Acuratețe = 0.3894

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	60	87	24	45	90	29	42	101	92
1:	16	238	24	43	14	20	24	104	44
2:	20	64	146	27	45	97	33	80	21
3:	18	58	9	303	33	66	23	45	23
4:	19	50	41	38	180	72	20	106	28
5:	6	15	31	56	37	335	13	51	17
6:	45	65	11	48	90	29	147	69	76
7:	17	55	19	33	55	22	19	285	15
8:	48	51	15	48	49	43	34	36	253

Am observat că modelul cu 12 intervale obține cea mai bună acuratețe și am trimis o încercare pe datele de testare pe site-ul competiției și am obținut o acuratețe de 0.36986.

2. MLPClassifier

Înainte de toate, am normalizat datele de antrenare și datele de validare folosind clasa *StandardScaler* din *sklearn.preprocessing* și metodele *fit_transform* (pentru mulțimea de antrenare), respectiv *transform* (pentru mulțimea de validare) ale aceleiași clase. Apoi, am instanțiat un model *MLPClassifier* cu diferiți hiperparametri, am antrenat modelul pe mulțimea train folosind metoda *fit* și am extras etichetele prezise și acuratețea cu care au fost prezise folosind metoda *predict*, respectiv funcția *accuracy_score* (din *sklearn.metrics*). Am reprezentat matricea de confuzie pentru fiecare model încercat. Rezultatele arată astfel:

- Funcție de activare='tanh', Hidden Layer Sizes = (1), Learning Rate = 0.01, Momentum = 0, Nr. Maxim de Iterații = 200 :

Acuratete = 0.1844

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	0	0	0	0	441	126	3	0	0
1:	0	0	3	0	431	93	0	0	0
2:	0	0	0	0	403	129	1	0	0
3:	0	0	1	0	141	435	1	0	0
4:	0	0	1	0	451	101	0	0	1
5:	0	0	0	0	90	465	4	0	2
6:	0	0	8	0	168	400	3	0	1
7:	0	0	0	0	378	140	2	0	0
8:	0	0	3	0	83	487	1	0	3

- Funcție de activare='tanh', Hidden Layer Sizes = (10), Learning Rate = 0.01, Momentum = 0, Nr. Maxim de Iterații = 200 :

Acuratete = 0.5384

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	211	52	25	36	69	11	43	69	54
1:	39	308	28	16	30	8	32	49	17
2:	24	49	290	14	45	42	24	28	17
3:	34	33	20	307	23	44	28	50	39
4:	70	34	49	37	238	36	24	49	17
5:	10	4	36	37	15	389	10	45	15
6:	40	35	24	20	10	15	367	12	57
7:	41	32	29	42	34	34	20	278	10
8:	40	15	7	33	7	30	116	25	304

- Funcție de activare='tanh', Hidden Layer Sizes = (10), Learning Rate = 0.00001, Momentum = 0, Nr. Maxim de Iterații = 200 :

Acuratete = 0.5394

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	180	79	43	23	77	6	47	56	59
1:	36	258	23	21	18	12	49	65	45
2:	38	21	289	22	48	37	40	27	11
3:	15	36	16	312	38	72	31	23	35
4:	57	22	55	34	266	15	22	52	31
5:	3	8	28	42	15	371	10	64	20
6:	17	42	18	13	18	14	388	19	51
7:	31	32	20	35	51	45	23	271	12
8:	25	42	7	15	11	37	69	9	362

- Funcție de activare='tanh', Hidden Layer Sizes = (10), Learning Rate = 10, Momentum = 0, Nr. Maxim de Iterații = 200 :

Acuratete = 0.1468

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	2	16	142	65	41	60	216	16	12
1:	2	24	39	102	59	58	221	11	11
2:	1	26	82	52	60	99	183	18	12
3:	1	8	88	25	75	65	299	17	0
4:	3	8	139	29	54	112	159	45	5
5:	0	3	65	16	71	86	253	62	5
6:	0	9	59	31	35	5	426	11	4
7:	1	10	62	108	14	58	222	34	11
8:	0	5	139	8	31	13	361	19	1

- Funcție de activare='tanh', Hidden Layer Sizes = (10), Learning Rate = 0.01, Momentum = 0, Nr. Maxim de Iterații = 20 :

Acuratete = 0.5004

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	173	56	45	26	83	11	68	50	58
1:	49	293	48	21	20	8	28	37	23
2:	20	45	318	17	37	40	25	24	7
3:	33	35	16	319	36	39	34	41	25
4:	82	64	78	50	179	23	27	35	16
5:	5	9	46	54	21	351	6	47	22
6:	100	27	29	35	13	17	271	31	57
7:	50	36	19	35	45	27	26	268	14
8:	39	24	7	42	16	34	67	18	330

- Funcție de activare='tanh', Hidden Layer Sizes = (10, 10), Learning Rate = 0.01, Momentum = 0, Nr. Maxim de Iterații = 2000 :

Acuratete = 0.5204

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	183	69	25	27	79	16	24	59	88
1:	42	318	38	18	18	3	17	45	28
2:	14	63	276	25	61	31	29	16	18
3:	31	32	18	339	39	48	12	24	35
4:	52	55	53	44	260	25	10	34	21
5:	3	11	46	49	28	349	9	43	23
6:	51	37	49	28	17	19	294	17	68
7:	30	50	27	38	51	42	19	241	22
8:	41	31	8	23	6	23	72	31	342

- Funcție de activare='relu', Hidden Layer Sizes = (10, 10), Learning Rate = 0.01, Momentum = 0, Nr. Maxim de Iterații = 2000 :

Acuratete = 0.6274

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	265	59	24	18	20	4	56	45	79
1:	27	357	21	13	8	10	36	40	15
2:	16	46	348	16	36	25	18	17	11
3:	32	27	23	345	20	30	29	33	39
4:	82	65	40	35	222	18	16	54	22
5:	11	9	26	49	17	382	11	36	20
6:	23	13	11	9	2	4	449	6	63
7:	47	29	18	28	12	20	24	324	18
8:	30	10	6	10	3	7	52	14	445

- Funcție de activare='relu', Hidden Layer Sizes = (100, 100), Learning Rate = 0.01, Momentum = 0, Nr. Maxim de Iterații = 2000 :

Acuratete = 0.6544

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	253	38	19	4	85	16	42	17	96
1:	12	425	14	8	17	9	11	18	13
2:	8	32	341	5	59	67	9	5	7
3:	11	20	22	289	94	63	32	7	40
4:	41	35	41	13	368	19	3	20	14
5:	1	4	11	9	18	472	7	22	17
6:	19	18	6	2	4	11	462	2	56
7:	24	59	6	15	59	119	15	194	29
8:	26	3	3	6	3	23	42	3	468

3. Rețea neuronală convoluțională

Înainte de toate, am normalizat datele de antrenare și datele de validare folosind clasa *StandardScaler* din *sklearn.preprocessing* și metoda *transform* a clasei, pentru ambele mulțimi de date(atât train, cât și validation).

Am creat o stivă cu mai multe straturi pentru model:

- 1 strat de input cu mărimea $32 \times 32 \times 1$ deoarece imaginile sunt alb-negru cu mărimea de 32×32 .
- 1 strat convoluțional 2D cu 16 filtre, mărimea kernel-ului 3×3 , funcția de activare 'relu'
- 1 strat ce realizează operația de max pooling pentru date 2D cu pool size de 2×2 .
- 1 strat convoluțional 2D cu 32 filtre, mărimea kernel-ului 3×3 , funcția de activare 'relu'
- 1 strat ce realizează operația de max pooling pentru date 2D cu pool size de 2×2 .
- 1 strat convoluțional 2D cu 64 filtre, mărimea kernel-ului 3×3 , funcția de activare 'relu'
- 1 strat ce realizează dropout pentru input (pentru a reduce overfitting-ul)
- 1 strat ce realizează flatten pentru input
- 1 strat dens (cu 10 unități și funcție de activare *softmax*) pentru output

Am avut două abordări asupra straturilor convoluționale 2D, o abordare cu bias regulizor $L2(0.001)$ și una fără bias regulizor.

Am compilat modelul folosind un optimizer Adam cu diferite learning rate-uri și am antrenat modelul cu diferite batch size-uri și număr de epoci.

Rezultatele arată astfel:

- Model cu bias regulizor $L2(0.001)$ pe straturile convoluționale 2D, learning rate = 0.0003, batch size = 256, nr. Epoci = 32

Loss pe mulțimea de antrenare = 0.3577; Acuratețe pe mulțimea de antrenare = 0.8722

Loss pe mulțimea de validare = 0.4079; Acuratețe pe mulțimea de validare = 0.8610

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	470	7	6	9	35	2	8	17	16
1:	11	477	6	8	7	5	0	10	3
2:	6	12	436	12	25	27	2	12	1
3:	4	3	12	499	21	10	5	12	12
4:	24	10	21	20	445	9	1	15	9
5:	3	1	9	20	9	503	5	8	3
6:	18	4	6	3	4	6	522	7	10
7:	12	13	12	18	9	18	3	432	3
8:	22	1	0	5	6	7	14	1	521

- Model cu bias regulizer L2(0.001) pe straturile convolutionale 2D, learning rate = 0.0003, batch size = 64, nr. Epoci = 32

Loss pe mulțimea de antrenare = 0.2770; Acuratețe pe mulțimea de antrenare = 0.9002

Loss pe mulțimea de validare = 0.3890; Acuratețe pe mulțimea de validare = 0.8668

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	467	6	9	8	26	0	17	19	18
1:	10	461	14	9	5	4	2	20	2
2:	5	2	469	8	12	19	3	14	1
3:	6	1	18	494	15	7	10	15	12
4:	25	5	36	23	432	9	1	17	6
5:	1	1	13	18	4	493	8	21	2
6:	11	3	3	1	1	4	549	2	6
7:	9	3	17	18	4	9	5	453	2
8:	19	4	0	10	3	3	18	4	516

- Model fara bias regulizer pe straturile convolutionale 2D, learning rate = 0.003, batch size = 256, nr. Epoci = 8

Loss pe mulțimea de antrenare = 0.4159; Acuratețe pe mulțimea de antrenare = 0.8538

Loss pe mulțimea de validare = 0.4606; Acuratețe pe mulțimea de validare = 0.8386

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	426	17	16	13	36	5	15	14	28
1:	5	471	18	12	5	3	1	11	1
2:	3	7	463	9	17	22	4	8	0
3:	2	4	20	476	13	24	11	11	17
4:	20	10	40	29	418	13	2	15	7
5:	2	1	17	16	8	493	8	15	1
6:	14	4	6	3	2	8	534	3	6
7:	19	10	22	21	14	19	12	402	1
8:	18	2	0	5	5	7	26	4	510

- Model fara bias regulizer pe straturile convolutionale 2D, learning rate = 0.003, batch size = 128, nr. Epoci = 8

Loss pe mulțimea de antrenare = 0.4132; Acuratețe pe mulțimea de antrenare = 0.8494

Loss pe mulțimea de validare = 0.4507; Acuratețe pe mulțimea de validare = 0.8424

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	466	12	6	7	37	1	9	16	16
1:	12	468	10	8	11	3	3	10	2
2:	5	11	455	6	26	18	3	9	0
3:	8	3	21	464	32	12	7	17	14
4:	29	10	28	15	448	2	1	14	7
5:	7	3	19	18	15	472	9	16	2
6:	18	4	5	2	0	3	535	3	10
7:	24	15	17	10	20	10	9	412	3
8:	31	4	0	10	5	5	29	1	492

- Model fara bias regulizer pe straturile convolutionale 2D, learning rate = 0.003, batch size = 64, nr. Epoci = 8

Loss pe mulțimea de antrenare = 0.4305; Acuratețe pe mulțimea de antrenare = 0.8475

Loss pe mulțimea de validare = 0.5111; Acuratețe pe mulțimea de validare = 0.8170

Matrice de confuzie:

	0	1	2	3	4	5	6	7	8
0:	423	20	12	13	32	2	13	23	32
1:	13	462	12	8	4	3	2	19	4
2:	4	15	427	8	27	25	6	19	2
3:	4	6	16	465	22	10	10	24	21
4:	26	17	31	27	408	10	4	22	9
5:	10	2	16	23	11	452	16	26	5
6:	11	14	3	2	5	1	523	6	15
7:	12	13	16	20	10	15	11	421	2
8:	18	8	1	9	2	4	26	5	504

Am observat că pe modelul cu bias regulizor L2(0.001) pe straturile convoluționale 2D, learning rate = 0.0003, batch size = 64, nr. Epoci = 32 se obține cea mai bună acuratețe, am trimis o încercare pe datele de testare pe site-ul competiției și am obținut o acuratețe de 0.85680.