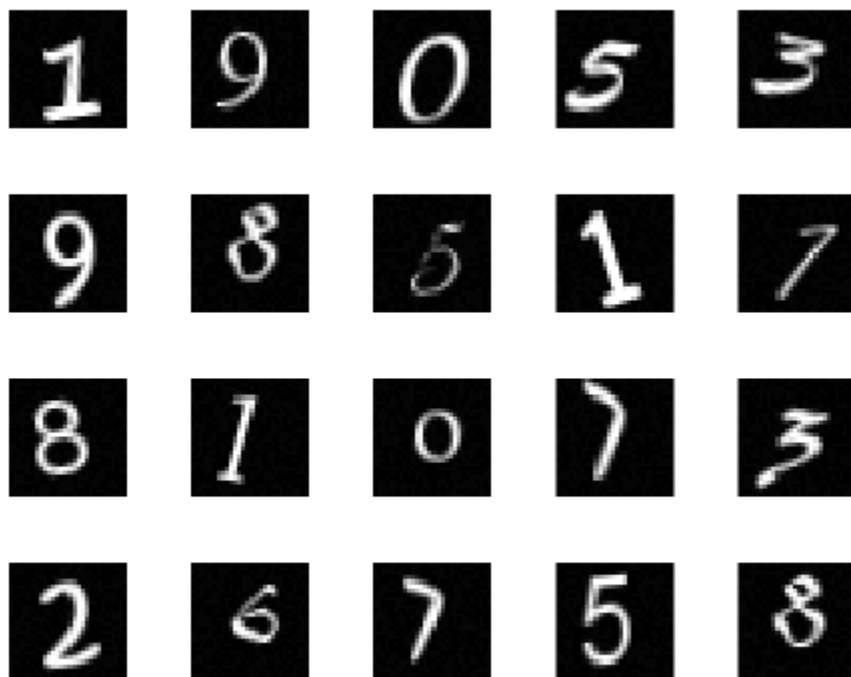


Retea neurala pentru clasificarea cifrelor Proiect Inteligenta Artificiala

1. Prezentare cerinta si setul de date

Acest proiect consta in utilizarea unei retele neurale pentru a clasifica un set de 5000 de imagini, fiecare imagine reprezentand una dintre cifrele din multimea $D = \{ 0, 1, \dots, 9 \}$.

Pentru a putea clasifica structuri complexe precum imaginile, reteaua noastra va trebui sa contina mai mult de un singur strat. In exemplul de fata avem imagini a caror dimensiune sunt de 28x28 pixeli, ceea ce inseamna un total de 784 de pixeli pentru fiecare imagine in parte.



De asemenea avem la dispozitie o multime de antrenare de dimensiune 784x1000, asta inseamna 1000 de imagini, dar si o multime tinta de dimensiune 1x1000, fiecare element reprezentand un numar de la 1 la 10, fiecare cifra reprezentand cifra reprezentata de imagine, iar pentru valoarea 10 cifra reprezentata va fi 0.

Complexitatea clasificarii consta in diferenta de reprezentare a cifrelor dupa cum se poate observa in imaginea de mai sus. Se doreste ca reteaua sa poata recunoaste o raza cat mai diversa de reprezentare a cifrelor (0 poate fi un oval sau chiar un cerc, 3 sa aiba o codita, 7 sa se asemeze cu 1, etc).

2. Structura rețelei și prelucrarea setului de date

Dupa cum am prezentat adineauri, rețeaua noastră va avea mai mult de un singur strat, și anume un strat ascuns și unul pentru afisarea rezultatului.

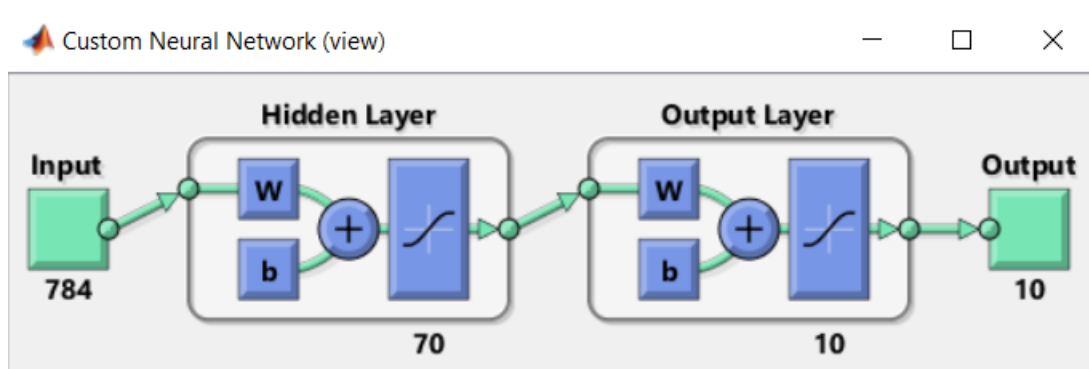
Vom învăța rețeaua să clasifice mulțimea de antrenare (784×1000) cu mulțimea tinta (1×1000), după care o simulăm pentru a clasifica cele 5000 de imagini de test (784×5000).

Având o mulțime tinta a cărei valori se află în raza 1-10, va fi dificil pentru o rețea să facă o clasificare directă a acestor valori astfel încât să avem o rată de învățare cât mai mare. Așadar vom codifica fiecare cifră printr-un bloc de 10 elemente, fiecare element având valoarea 1 sau 0. Dacă poziția i conține 1, atunci valoarea tinta este i .

Ideea de rezolvare a acestei probleme este destul de simplă: vom lua fiecare imagine și pentru stratul ascuns al rețelei vom face o compresie a acesteia prin intermediul a 70 de perceptroni. Pur și simplu vom învăța rețeaua noastră să codifice o imagine folosind doar 70 de valori în loc de 784. Cum mulțimea tinta este de dimensiune 10×1000 asta înseamnă că ultimul strat al rețelei va conține exact 10 perceptroni, așadar va fi mai ușor pentru rețea să clasifice în 10 moduri elemente a câte 70 de valori fiecare, decât a câte 784.

De ce 70 de perceptroni? Pentru că un număr mai mic de perceptroni ar duce la pierderea de date, iar un număr mai mare ar îngreuna ultimul strat pentru clasificarea în cele 10 moduri. Așadar performanța ar avea de suferit ori din cauza primului strat pentru număr mic de perceptroni, ori din cauza ultimului strat în cazul unui număr mare de perceptroni.

Ce funcții vom folosi? Având o mulțime tinta cu valori 0 sau 1, vom avea nevoie de funcție de transfer care returnează valori aparținând multimii $\{0, 1\}$ pentru a nu exista valori de 1 în mai mult de o poziție pentru fiecare bloc. Așadar funcția 'hardlim' nu are sens să o abordăm. Deoarece dorim o valoare strict pozitivă dar care să nu atingă valoarea 1, vom recurge la funcția de transfer 'tansig', iar ca funcție de antrenare vom folosi metoda "Scaled conjugate gradient backpropagation", regăsită în matlab sub denumirea 'trainscg'.



3. Verificare eficienta retea si platforma Kaggle

Pentru a putea verifica eficienta retelei, avem la dispozitie platforma Kaggle care verifica rezultatele noastre si de asemenea calculeaza procentual datele clasificate corect de catre retea. Reteaua prezentata anterior obtine un punctaj de 0.89

57	↓7	Ana-Maria Bucur	0.89200	6
58	↓2	Maria-Petre	0.89200	2
59	↓1	Irina Botez	0.89200	4
60	↓1	GaneaDanAndrei	0.89125	6
61	↑14	Robert Manea	0.89075	8
-		AlexandruRusu	0.89000	-
Post-Deadline Entry If you would have submitted this entry during the competition, you would				
62	↑6	NiculescuBogdan	0.88975	2
63	↑4	IulianaGeorgescu	0.88950	17
64	↑5	Istratie Vladut-Andrei	0.88900	4
65	↑1	RubenBacanu	0.88875	7

Deoarece platforma asteapta ca rezultat o structura de forma 1x5000, va trebui ca pentru fiecare bloc rezultat din retea sa alegem valoarea maxima, deoarece functia tansig va returna valori cuprinse in intervalul (0,1) si in locul maximului vom avea valoarea 1, iar in rest 0.

Fisierul de testare pentru platforma kaggle va fi un fisier de tip csv cu 2 coloane "Id" si "Prediction", iar "Prediction" va contine indicele din fiecare bloc care contine valoarea 1.

DELIMITERS		
rezultat.csv		
	A	B
	Id	Prediction
	NUMBER	NUMBER
1	Id	Prediction
2	1	6
3	2	2
4	3	2
5	4	4
6	5	7
7	6	3
8	7	7
9	8	4
10	9	4
11	10	2
12	11	4
13	12	5
14	13	6

4. Definirea rețelei in cod Matlab

```
hiddenSize1 = 70;  
  
autoenc1 = newff(X, T, [hiddenSize1], {'tansig' 'tansig'}, 'trainscg');  
  
autoenc1.trainParam.epochs = 400;  
autoenc1.trainParam.goal = 0.001;  
  
autoenc1.divideParam.trainRatio = 70/100;  
autoenc1.divideParam.valRatio = 15/100;  
autoenc1.divideParam.testRatio = 15/100;  
  
[res1 output] = train(autoenc1, X, T);  
  
Y = sim(res1, X);  
  
resultKaggle = sim(res1, Xkaggle);
```