Tema 2 Invățare Automată

Văideanu Renata - Georgia

January 12, 2025

Abstract

Tema 2 la materia Învațare Automata, anul 4 - Universitatea Națională de Știință și Tehnologie Politehnică București

Introducere

Acest document explică procesul de implementare a sarcinilor din Tema 2 de Învățare Automată, cu accent pe clasificarea imaginilor din două seturi de date diferite: Fashion-MNIST și Fruits-360. Obiectivul principal este proiectarea unor arhitecturi de rețele neurale care să efectueze automat atât extragerea de atribute, cât și utilizarea lor în clasificare.

1 MLP pe atributele extrase în etapa 1

1.1 Refacerea pașilor din prima etapă

Am început procesul prin implementarea etapelor specifice primei părți a temei, concentrându-mă pe standardizarea și selecția atributelor. Pentru standardizarea datelor și reducerea dimensionalității, am utilizat metodele HOG (Histogram of Oriented Gradients) și PCA (Principal Component Analysis), asigurând o reprezentare optimizată a caracteristicilor relevante.

În ceea ce privește selecția atributelor pentru cele două seturi de date, am optat pentru metoda SelectPercentile, care permite păstrarea unui anumit procentaj al celor mai semnificative caracteristici. Am stabilit valoarea procentajului la 10% pentru setul de date Fruits-360, datorită complexității și diversității moderate a atributelor, și la 30% pentru setul de date Fashion-MNIST, având în vedere natura mai complexă și detaliată a informațiilor necesare pentru clasificare.

Fruits:

- Original feature shape for training: (70491, 920)
- Select Percentile shape for training: (70491, 92)
- Original feature shape for test: (23619, 920)
- Select Percentile shape for test: (23619, 92)

Fashion:

- Original feature shape for training: (60000, 164)
- Select Percentile shape for training: (60000, 49)
- Original feature shape for test: (10000, 164)
- Select Percentile shape for test: (10000, 49)

1.2 Aplicarea MLP

Modelul MLP (Multi-Layer Perceptron) este o rețea neuronală feed-forward implementată în PyTorch, concepută pentru probleme de clasificare. Modelul pe care l-am ales este compus din trei straturi complet conectate (dense): primul transformă datele de intrare (input_size) într-un spațiu latent de 128 dimensiuni, al doilea menține această dimensiune pentru a aprofunda învățarea, iar ultimul produce scorurile pentru clasificare (num_classes). Între primele două straturi, se aplică funcția de activare ReLU pentru a introduce non-liniaritate și a captura relații complexe în date.

Preprocesarea datelor

Se utilizează StandardScaler pentru a aduce caracteristicile setului de date la o distribuție normală, cu media 0 și deviația standard 1. Aceasta îmbunătățește performanța modelului, mai ales în cazul rețelelor neuronale, unde intrările normalizate accelerează convergența. Datele standardizate și etichetele sunt transformate în tensori PyTorch pentru a putea fi utilizate în rețea.

Configurarea modelului și a antrenării

Dimensiunile modelului:

- input_size: Dimensiunea caracteristicilor de intrare este egală cu numărul de atribute după preprocesare.
- output_size: Este egal cu numărul de clase din set: Fruits-360 (141) și Fashion-MNIST (10).

Hiperparametri:

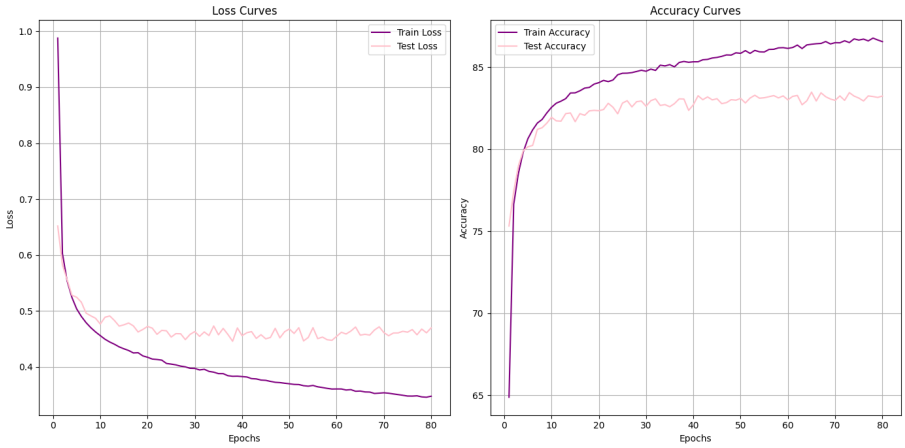
- batch_size: Batch size-ul mai mare (128) pentru Fruits-360 este potrivit pentru a gestiona complexitatea datelor și stabilitatea gradientului, în timp ce pentru Fashion-MNIST, un batch size mai mic (64) permite actualizări frecvente și mai adaptate pentru un set de date simplu.
- learning_rate: Learning rate-ul mai mare (5e-5) pentru Fruits-360 este ales pentru a accelera antrenarea pe un set mare și complex, în timp ce un learning rate mai mic (1e-5) pentru Fashion-MNIST ajută la ajustări fine, evitând supraconvergența pe un set mai mic și mai simplu.
- num_epochs = 80: Numărul de epoci este potrivit pentru convergență, având în vedere dimensiunea seturilor folosite.

Pentru setul de date Fruits-360, am inclus un factor de Weight Decay (4e-3) în cadrul optimizerului. Această penalizare aplicată greutăților contribuie la prevenirea overfitting-ului, fiind esențială pentru un set de date cu un număr mare de clase și o posibilă redundanță între exemple.

Aceste grafice conțin informații importante despre evoluția pierderii (loss) și acurateței (accuracy) pentru seturile de date Fashion-MNIST și Fruits-360 în timpul antrenării modelului. Iată o analiză pentru fiecare set de date, pe baza observațiilor:

1. Fashion-MNIST

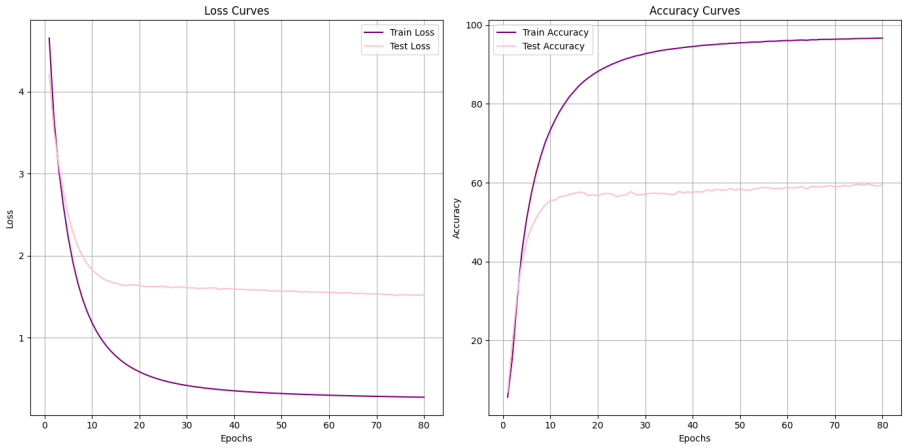
Pierdere (Loss): Pierderea pe setul de antrenare scade constant, indicând că modelul învață caracteristicile relevante. Pierderea pe setul de testare scade inițial, dar după un punct devine mai stabilă, cu fluctuații ușoare. Această stabilizare a pierderii pe setul de testare sugerează că modelul se apropie de capacitatea sa optimă de generalizare pentru acest set de date.
Acuratețe (Accuracy): Acuratețea pe setul de antrenare crește constant și ajunge aproape de 90%. Pe setul de testare, acuratețea este mai mică, atingând aproximativ 85% spre sfârșitul antrenării. Diferența dintre acuratețea pe antrenare și testare este mică, ceea ce indică faptul că modelul nu suferă de overfitting semnificativ.



Concluzie: Modelul are o performanță bună, iar hiperparametrii sunt adecvați pentru acest set de date. Fluctuațiile din pierderea și acuratețea pe test sugerează un echilibru între învățare și generalizare.

2. Fruits-360

Pierdere (Loss): Pierderea pe setul de antrenare scade rapid și consistent, iar pe setul de testare, scăderea este mai lentă, stabilizându-se după aproximativ 20 de epoci. Stabilizarea pierderii pe test la un nivel relativ mai mare sugerează că modelul întâmpină dificultăți în a generaliza pe acest set mai complex.
Acuratețe (Accuracy): Acuratețea pe setul de antrenare crește rapid, depășind 95%, ceea ce arată că modelul învață bine pe setul de antrenare. Pe setul de testare, acuratețea crește mai lent și se stabilizează în jurul valorii de 65%-70%. Diferența notabilă între acuratețea pe antrenare și testare indică un posibil overfitting, ceea ce poate fi explicat prin complexitatea ridicată a setului de date (141 clase și imagini cu variabilitate ridicată).



Concluzie: Deși modelul învață bine pe setul de antrenare, performanța pe testare este limitată. Acest lucru sugerează că s-ar putea explora metode adiționale pentru prevenirea overfitting-ului, cum ar fi regularizarea mai puternică, augmentarea datelor sau ajustarea hiperparametrilor.

Comparație generală: Fashion-MNIST: Modelul generalizează bine, cu o diferență mică între pierderea și acuratețea pe seturile de antrenare și testare. Este potrivit pentru un set cu 10 clase și variabilitate mai mică. Fruits-360: Modelul înregistrează un overfitting moderat, datorită complexității mai ridicate a setului de date. Se pot face ajustări pentru a îmbunătăți generalizarea. Pe baza graficelor, Fashion-MNIST este mai ușor de gestionat, în timp ce Fruits-360 necesită strategii suplimentare pentru a îmbunătăți performanța.

2 Arhitectura de tip MLP direct peste imagini

Modelul MLP (Multi-Layer Perceptron) este acelasi ca la prima cerință. Modelul pe care l-am ales este compus din trei straturi complet conectate (dense): primul transformă datele de intrare (input_size) într-un spațiu latent de 128 dimensiuni, al doilea menține această dimensiune pentru a aprofunda învățarea, iar ultimul produce scorurile pentru clasificare (num_classes). Între primele două straturi, se aplică funcția de activare ReLU pentru a introduce non-liniaritate și a captura relații complexe în date.

Preprocesarea datelor

Se utilizează StandardScaler pentru a aduce caracteristicile setului de date la o distribuție normală, cu media 0 și deviația standard 1. Aceasta îmbunătățește performanța modelului, mai ales în cazul rețelelor neuronale, unde intrările normalizate accelerează convergența. Datele standardizate și etichetele sunt transformate în tensori PyTorch pentru a putea fi utilizate în rețea.

Configurarea modelului și a antrenării

Dimensiunile modelului:

- input_size: Dimensiunea caracteristicilor de intrare este egală cu numărul de atribute după preprocesare.
- output_size: Este egal cu numărul de clase din set: Fruits-360 (141) și Fashion-MNIST (10).

Hiperparametri:

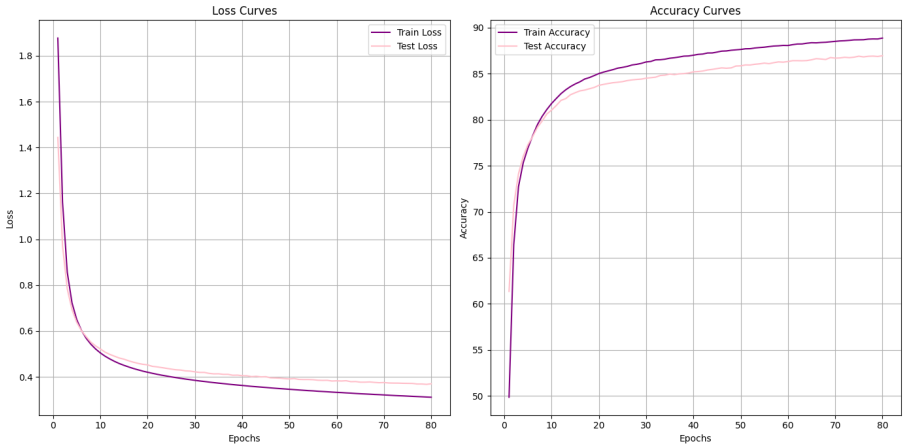
- batch_size = 64: Batch size-ul este potrivit pentru a gestiona complexitatea datelor și stabilitatea gradientului, potrivit pentru actualizări frecvente.
- learning_rate: Learning rate-ul mai mare (1e-4) pentru Fruits-360 ajută modelul să exploreze mai mult spațiul de parametri și să evite blocarea într-un minim local, ceea ce este esențial pentru datele cu multe clase, în timp ce un learning rate mai mic (1e-5) pentru Fashion-MNIST permite o convergență mai precisă, importantă pentru un set de date mai puțin variat, reducând riscul de overfitting.
- num_epochs = 80: Numărul de epoci este potrivit pentru convergență, având în vedere dimensiunea seturilor folosite.

Pentru setul de date Fruits-360, imaginile au fost redimensionate de la dimensiunea originală de 100x100 pixeli la 64x64 pixeli, pentru a facilita procesarea acestora de către model și pentru a reduce cerințele de calcul.

Aceste grafice conțin informații importante despre evoluția pierderii (loss) și acurateței (accuracy) pentru seturile de date Fashion-MNIST și Fruits-360 în timpul antrenării modelului. Iată o analiză pentru fiecare set de date, pe baza observațiilor:

1. Fashion-MNIST

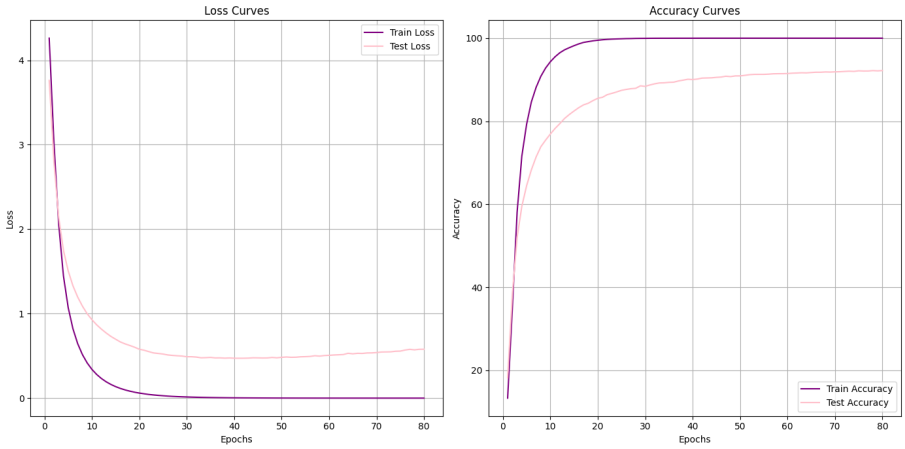
Pierdere (Loss): Pierderea pe setul de antrenare scade constant, indicând că modelul învață caracteristicile relevante. Pierderea pe setul de testare scade în aceeasi manieră ca în cazul setului de train, scăzând chiar sub 0.4 dupa 50 de epoci. Această stabilizare a pierderii pe setul de testare sugerează că modelul se apropie de capacitatea sa optimă de generalizare pentru acest set de date.
Acuratețe (Accuracy): Acuratețea pe setul de antrenare crește constant și ajunge aproape de 90%. Pe setul de testare, acuratețea este puțin mai mică, trecând peste 85% spre sfârșitul antrenării. Diferența dintre acuratețea pe antrenare și testare este mică, ceea ce indică faptul că modelul nu suferă de overfitting semnificativ.



Concluzie: Modelul are o performanță bună, iar hiperparametrii sunt adecvați pentru acest set de date. Fluctuațiile din pierderea și acuratețea pe test sugerează un echilibru între învățare și generalizare.

2. Fruits-360

Pierdere (Loss): Pierderea pe setul de antrenare scade rapid și consistent, iar pe setul de testare, scăderea este mai lentă, stabilizându-se după aproximativ 20 de epoci. Stabilizarea pierderii pe test la un nivel puțin mai mare sugerează că modelul întâmpină mici dificultăți în a generaliza pe acest set mai complex.
Acuratețe (Accuracy): Acuratețea pe setul de antrenare crește rapid, depășind atingând chiar 100%, ceea ce arată că modelul învață bine pe setul de antrenare. Pe setul de testare, acuratețea crește și se stabilizează dupa mai multe epoci, în jurul valorii de 90%. Diferența dintre acuratețea pe antrenare și testare indică că modelul poate suferi într-o mica masură de overfitting, ceea ce poate fi explicat prin complexitatea ridicată a setului de date (141 clase și imagini cu variabilitate ridicată).



Concluzie: Se poate observa că modelul învață bine pe setul de antrenare, cu o performanță usor mai scăzută pe partea de testare.

Comparație generală: Fashion-MNIST: Modelul generalizează bine, cu o diferență mică între pierderea și acuratețea pe seturile de antrenare și testare. Este potrivit pentru un set cu 10 clase și variabilitate mai mică. Fruits-360: Modelul generalizează mai puțin bine, datorită complexității mai ridicate a setului de date. Pe baza graficelor, Fashion-MNIST este mai ușor de gestionat, în timp ce Fruits-360 necesită strategii suplimentare pentru a îmbunătăți performanța.

3 Arhitectura de tip convoluțional

Rețeaua neuronală convoluțională (CNN) utilizează PyTorch și torchvision, având o arhitectură care include straturi convoluționale: conv1 cu 32 de filtre 3x3 pentru intrări cu num_channels, conv2 cu 64 de filtre 3x3 pentru 32 de canale, conv3 cu 128 de filtre 3x3 și stride=2 pentru downsampling, urmate de Batch Normalization (bn1, bn2, bn3) și activare ReLU după fiecare strat convoluțional. După convoluții, se aplică Global Average Pooling pentru a reduce dimensiunea spațială la o valoare per canal, iar stratul complet conectat (FC) include Linear(128, 64) pentru a proiecta cele 128 de caracteristici într-un vector de 64, Dropout pentru regularizare și Linear(64, num_classes) pentru clasificare în num_classes clase. Preprocesarea datelor presupune augmentare cu întoarcere aleatorie, rotație și normalizare (mean=0.5, std=0.5) sau doar normalizare în absența augmentării. Codul este modular și flexibil, permițând personalizarea parametrilor și oferind opțiuni pentru regularizare și augmentare, fiind ideal pentru clasificarea imaginilor.

Preprocesarea datelor

În cazul Fashion-MNIST am transformat datele de intrare în tensori PyTorch și am adaugat o dimensiune suplimentară pentru canal (1), necesară pentru a le face compatibile cu o rețea CNN.

Configurarea modelului și a antrenării

Dimensiunile modelului:

- input_size: Dimensiunea caracteristicilor de intrare este egală cu numărul de atribute după preprocesare.
- output_size: Este egal cu numărul de clase din set: Fruits-360 (141) și Fashion-MNIST (10).

Hiperparametri:

- batch_size = 64: Batch size-ul este potrivit pentru a gestiona complexitatea datelor și stabilitatea gradientului, potrivit pentru actualizări frecvente.
- learning_rate: Learning rate-ul mai mare (1e-4) pentru Fruits-360 ajută modelul să exploreze mai mult spațiul de parametri și să evite blocarea într-un minim local, ceea ce este esențial pentru datele cu multe clase, în timp ce un learning rate mai mic (1e-5) pentru Fashion-MNIST permite o convergență mai precisă, importantă pentru un set de date mai puțin variat, reducând riscul de overfitting.
- num_epochs = 80: Numărul de epoci este potrivit pentru convergență, având în vedere dimensiunea seturilor folosite.

Pentru setul de date Fruits-360, imaginile au fost redimensionate de la dimensiunea originală de 100x100 pixeli la 16x16 pixeli, pentru a facilita procesarea acestora de către model și pentru a reduce cerințele de calcul.

Aceste grafice conțin informații importante despre evoluția pierderii (loss) și acurateței (accuracy) pentru seturile de date Fashion-MNIST și Fruits-360 în timpul antrenării modelului. Iată o analiză pentru fiecare set de date, pe baza observațiilor:

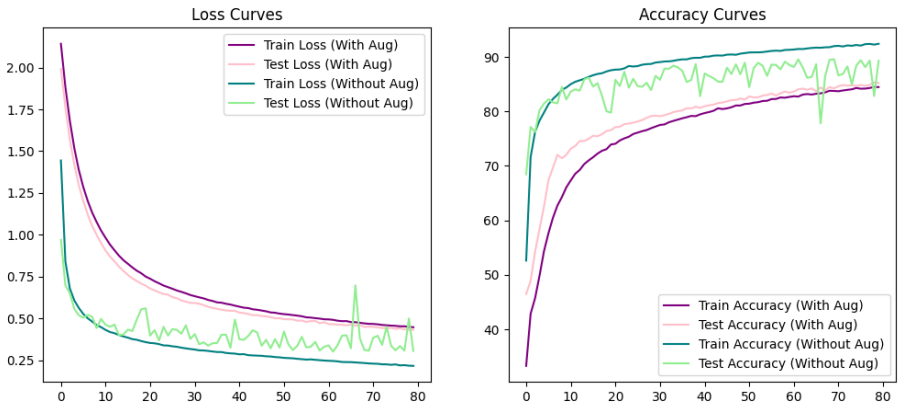
1. Fashion-MNIST

Pierdere (Loss):

- **Cu augmentare:** Pierderea scade gradual atât pentru setul de antrenare, cât și pentru cel de testare, ceea ce indică o convergență stabilă. Valoarea finală este mică, arătând o bună generalizare.
- **Fără augmentare:** Pierderea este mai mică în faza de antrenare, dar mai variabilă pentru setul de testare, ceea ce sugerează un risc mai ridicat de overfitting.

Acuratețe (Accuracy):

- **Cu augmentare:** Acuratețea crește progresiv, atingând valori apropiate pentru seturile de antrenare și testare. Aceasta arată că augmentarea contribuie la o mai bună generalizare și stabilitate.
- **Fără augmentare:** Acuratețea pentru antrenare este mai mare decât cea pentru testare, ceea ce indică o posibilă supraînvățare a modelului pe datele de antrenare.



Concluzie:
Modelul cu augmentare oferă o performanță mai robustă, reducând riscul de overfitting și îmbunătățind generalizarea. În schimb, modelul fără augmentare prezintă o convergență rapidă, dar mai puțin stabilă pe setul de testare, fiind potrivit pentru seturi de date cu variabilitate redusă și un număr mic de clase. Fără augmentare, modelul tinde să supraînvețe datele de antrenare, dar performanța generală rămâne stabilă. Setul Fashion-MNIST este mai ușor de gestionat și antrenat comparativ cu alte seturi de date.

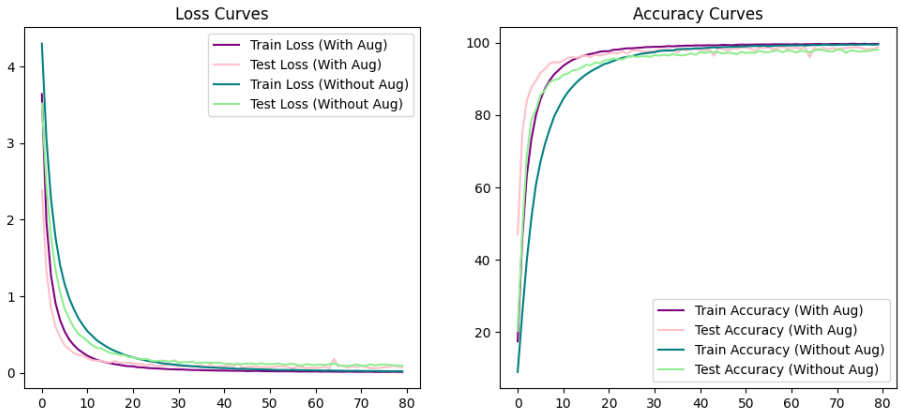
2. Fruits-360

Pierdere (Loss):

- **Cu augmentare:** Pierderea scade rapid la început și continuă să scadă constant până la valori aproape zero pentru antrenare și testare, indicând o convergență foarte bună.
- **Fără augmentare:** Pierderea urmează un comportament similar, dar convergența pare ușor mai rapidă decât în cazul augmentării, indicând o supraînvățare mai probabilă.

Acuratețe (Accuracy):

- **Cu augmentare:** Acuratețea atinge aproape 100% pentru seturile de antrenare și testare, demonstrând o generalizare excelentă.
- **Fără augmentare:** Deși acuratețea este foarte apropiată de cea cu augmentare, aceasta arată o ușoară tendință de overfitting, cu diferențe minore între seturile de antrenare și testare.



Concluzie: Atât modelul cu augmentare, cât și cel fără augmentare, ating performanțe ridicate, dar augmentarea adaugă robustețe și previne supraînvățarea pe datele de antrenare. Generalizarea este ușor afectată din cauza complexității datelor (141 de clase și imagini detaliate). Augmentarea contribuie semnificativ la prevenirea supraînvățării, oferind rezultate consistente între seturile de antrenare și testare. Fără augmentare, modelul tinde mai mult spre overfitting.

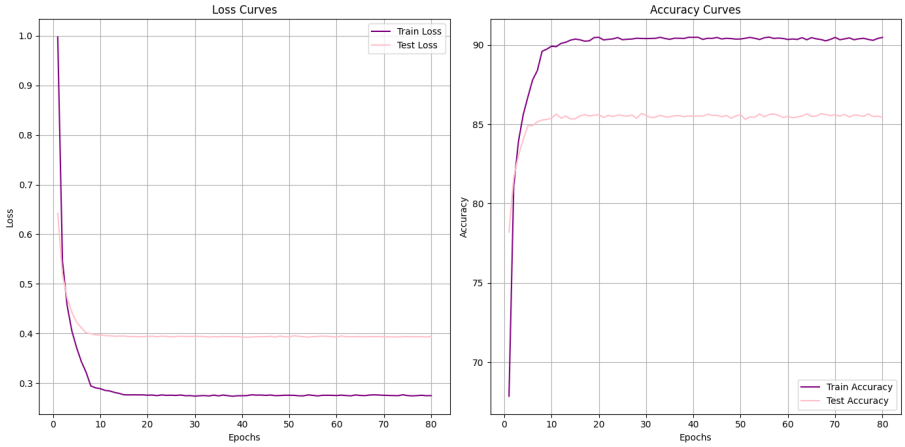
Comparație generală: Fashion-MNIST este mai ușor de gestionat datorită dimensiunii reduse și variabilității limitate, pe când Fruits-360 solicită mai multă atenție în configurarea modelului și strategii suplimentare pentru a aborda complexitatea ridicată a datelor. Graficele evidențiază importanța augmentării pentru ambele seturi, dar mai ales pentru seturile cu un număr mare de clase și variabilitate mai mare.

4 Utilizarea unei proceduri de finetuning peste arhitectura ResNet-18

Pentru această sarcină, s-a folosit o arhitectură pre-antrenată (e.g., ResNet-18) pe seturi de date standard, precum CIFAR-10, și s-a aplicat procedura de finetuning pentru a adapta modelul la seturile de date cerute: Fashion-MNIST și Fruits-360. Finetuning-ul presupune ajustarea ponderilor modelului pre-antrenat pentru sarcini noi, păstrând totodată caracteristicile de generalizare dobândite în pre-training.

1. Fashion-MNIST

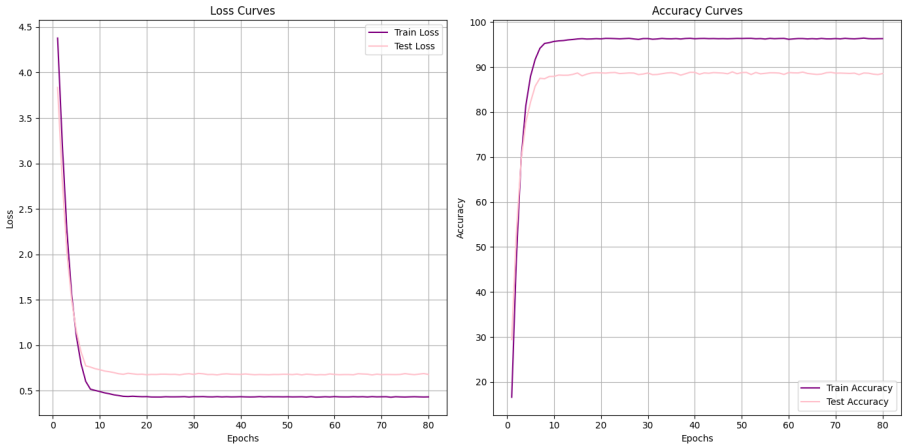
Pierdere (Loss): Diferența dintre pierderea pe setul de antrenare și cel de testare este mai mare, ceea ce sugerează o ușoară supraantrenare. Pierderea pe setul de testare scade sub 0.4, dar nu converg perfect cu pierderile de antrenare.
Acuratețe (Accuracy): Acuratețea pe setul de testare este ușor mai mică decât cea de pe setul de antrenare, dar se stabilizează după primele epoci. Modelul atinge o acuratețe de peste 85% pe setul de testare, evidențiind o diferență notabilă față de setul de antrenare.



Concluzie: Modelul arată o capacitate bună de învățare, dar diferența între seturile de antrenare și testare indică o dificultate în generalizare. Acest lucru sugerează că Fashion-MNIST, deși un set cu o variabilitate redusă, poate necesita o reglementare suplimentară pentru a reduce supraantrenarea.

2. Fruits-360

Pierdere (Loss): Diferența dintre pierderile de antrenare și testare este destul de mică, iar pierderea pe setul de testare scade constant, convergând aproape de cea de antrenare după 20 de epoci.
Acuratețe (Accuracy): Acuratețea pe setul de testare depășește 90%, ceea ce indică o performanță ridicată și o capacitate bună de generalizare, în ciuda diversității ridicate a datelor.



Concluzie: Fruits-360 beneficiază de strategia de finetuning utilizată, arătând o generalizare destul de bună. Modelul gestionează bine variabilitatea ridicată a datelor.

Comparație generală: Rezultatele obținute evidențiază diferențele în modul în care modelul ResNet-18, pre-antrenat pe CIFAR-10 și finetunat pentru sarcinile actuale, generalizează pe cele două seturi de date:

- Fashion-MNIST: Modelul demonstrează o bună capacitate de învățare, dar diferența mai mare între pierderile și acuratețile pe seturile de antrenare și testare sugerează o supraantrenare ușoară. Setul de date, deși mai simplu și cu o variabilitate redusă, necesită tehnici suplimentare de reglementare pentru a îmbunătăți generalizarea. Acest lucru arată că Fashion-MNIST este mai sensibil la finețea ajustării hiperparametrilor.
- Fruits-360: Modelul generalizează mai bine, atingând o acuratețe ridicată pe setul de testare. Complexitatea ridicată a setului de date este gestionată eficient, ceea ce indică faptul că strategia de finetuning a fost bine adaptată. Totuși, o ușoară îmbunătățire a performanței ar putea fi obținută prin optimizări adiționale.

În concluzie, Fashion-MNIST este un set de date mai ușor în termeni de complexitate, dar poate genera dificultăți în generalizare, iar Fruits-360, deși mai complex, permite o adaptare mai bună datorită variabilității bogate care ajută la prevenirea supraantrenării.

5 Raport de evaluare comparativ

Concluzii din etapa 1

Random Forest și Gradient Boosted Trees au obținut cea mai mare acuratețe pe setul Fruits-360 datorită capacității lor de a modela relații complexe între feature-uri, rezistenței la zgomot și outlieri, și concentrării pe cele mai relevante caracteristici. Random Forest oferă robustețe prin diversitatea arborilor săi, iar Gradient Boosted Trees optimizează iterativ performanța, gestionând eficient clase similare. Setul de date Fruits-360 a favorizat performanța acestor metode, care sunt ideale pentru date complexe cu relații non-lineare.

Setul de date Fashion-MNIST a obținut performanțe bune cu SVM și Gradient Boosted Trees datorită separabilității relativ bune între clase și dimensiunii moderate a datelor, care a permis ambelor metode să fie eficiente din punct de vedere computațional. SVM a reușit să separe eficient clasele utilizând kernel-ul adecvat, în timp ce Gradient Boosted Trees a optimizat erorile iterativ, având o performanță robustă, mai ales în fața unor clase vizual similare (de exemplu, tricouri și rochii). Astfel, ambele metode au gestionat bine variabilitatea și complexitatea setului de date.

5.1 Fruits-360

Random Forest

	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.84	1.00	0.91	157					
1	0.62	0.66	0.64	164					
2	0.77	0.84	0.80	148					
3	0.96	0.94	0.95	160					
4	0.66	0.95	0.78	164	Grape Blue 1	1.00	1.00	1.00	328
5	0.63	0.99	0.77	161	Plum 3	1.00	1.00	1.00	304
6	0.95	0.68	0.79	164	Melon Piel de Sapo 1	1.00	0.93	0.96	246
7	0.91	1.00	0.95	234	Tomato 3	0.98	0.90	0.94	246
8	0.77	0.70	0.74	152	Cherry Rainier 1	1.00	0.72	0.84	246
9	0.73	0.70	0.71	164	Cherry 2	0.92	0.99	0.96	246
10	0.73	0.76	0.74	164	Tomato 1	1.00	0.98	0.99	246
11	0.79	0.86	0.82	144	Strawberry Wedge 1	0.93	0.77	0.84	246
12	0.98	1.00	0.99	166	Peach 2	1.00	1.00	1.00	246
13	0.94	0.77	0.85	164	Walnut 1	1.00	1.00	1.00	249
14	0.85	0.98	0.91	219	micro avg	0.98	0.93	0.96	2603
15	0.94	0.94	0.94	164	macro avg	0.98	0.93	0.95	2603
16	0.79	0.97	0.87	143	weighted avg	0.98	0.93	0.96	2603
...	0.78	0.60	0.68	166					
accuracy			0.86	23619					
macro avg	0.87	0.86	0.86	23619					
weighted avg	0.90	0.89	0.89	23619					

GradientBoosted Trees

	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.89	1.00	0.94	157					
1	0.64	0.70	0.67	164					
2	0.88	0.72	0.79	148					
3	0.98	0.78	0.86	160					
4	0.79	0.83	0.81	164	Grape Blue 1	1.00	0.99	0.99	328
5	0.72	1.00	0.83	161	Plum 3	0.99	1.00	0.99	304
6	0.85	0.71	0.77	164	Melon Piel de Sapo 1	1.00	0.89	0.94	246
7	0.97	1.00	0.98	234	Tomato 3	0.98	0.89	0.93	246
8	0.73	0.83	0.78	152	Cherry Rainier 1	0.98	0.65	0.78	246
9	0.62	0.65	0.63	164	Cherry 2	1.00	0.98	0.99	246
10	0.87	0.90	0.88	164	Tomato 1	1.00	0.98	0.99	246
11	0.88	0.83	0.85	144	Strawberry Wedge 1	0.90	0.76	0.83	246
12	0.95	0.99	0.97	166	Peach 2	1.00	1.00	1.00	246
13	0.58	0.81	0.67	164	Walnut 1	1.00	1.00	1.00	249
14	0.92	0.98	0.95	219	micro avg	0.99	0.92	0.95	2603
15	0.98	0.70	0.82	164	macro avg	0.98	0.91	0.94	2603
16	0.82	0.90	0.85	143	weighted avg	0.98	0.92	0.95	2603
...									
accuracy			0.86	23619					
macro avg	0.87	0.86	0.86	23619					
weighted avg	0.87	0.86	0.86	23619					

Task 1

	precision	recall	f1-score	support
Apple 6	0.7044	0.7134	0.7089	157
Apple Braeburn 1	0.7895	0.1829	0.2970	164
Apple Crimson Snow 1	0.0698	0.0203	0.0314	148
Apple Golden 1	0.7880	0.9062	0.8430	160
Apple Golden 2	0.6822	0.9817	0.8050	164
Apple Golden 3	0.5610	0.2857	0.3786	161
Apple Granny Smith 1	0.9487	0.4512	0.6116	164
Apple Pink Lady 1	0.5987	0.8034	0.6861	234
Apple Red 1	0.2841	0.1645	0.2083	152
Apple Red 2	0.1720	0.0976	0.1245	164
Apple Red 3	0.7262	0.3720	0.4919	164
Apple Red Delicious 1	0.3431	0.5694	0.4282	144
Apple Red Yellow 1	0.3376	0.4819	0.3970	166
Apple Red Yellow 2	0.4641	0.5915	0.5201	164
Apple hit 1	0.9511	0.7991	0.8685	219
Apricot 1	0.8103	0.8598	0.8343	164
Avocado 1	0.3250	0.0909	0.1421	143
...				
accuracy			0.5884	23619
macro avg	0.6097	0.5865	0.5703	23619
weighted avg	0.6117	0.5884	0.5719	23619

Task 2

	precision	recall	f1-score	support
Apple 6	0.9401	1.0000	0.9691	157
Apple Braeburn 1	0.7595	0.7317	0.7453	164
Apple Crimson Snow 1	0.7500	0.9527	0.8393	148
Apple Golden 1	0.8377	1.0000	0.9117	160
Apple Golden 2	0.9787	0.8415	0.9049	164
Apple Golden 3	0.7318	1.0000	0.8451	161
Apple Granny Smith 1	1.0000	0.6951	0.8201	164
Apple Pink Lady 1	1.0000	1.0000	1.0000	234
Apple Red 1	0.8657	0.7632	0.8112	152
Apple Red 2	0.8990	0.5427	0.6768	164
Apple Red 3	0.9686	0.9390	0.9536	164
Apple Red Delicious 1	0.9820	0.7569	0.8549	144
Apple Red Yellow 1	1.0000	0.6386	0.7794	166
Apple Red Yellow 2	0.8956	0.9939	0.9422	164
Apple hit 1	0.9888	0.8037	0.8866	219
Apricot 1	1.0000	0.9146	0.9554	164
Avocado 1	0.9728	1.0000	0.9862	143
...				
accuracy			0.9129	23619
macro avg	0.9242	0.9082	0.9080	23619
weighted avg	0.9261	0.9129	0.9120	23619

Task 3

Cu augumentare

	precision	recall	f1-score	support
Apple 6	0.9691	1.0000	0.9843	157
Apple Braeburn 1	0.9320	0.8354	0.8810	164
Apple Crimson Snow 1	0.9500	0.8986	0.9236	148
Apple Golden 1	0.9748	0.9688	0.9718	160
Apple Golden 2	0.9879	0.9939	0.9909	164
Apple Golden 3	0.9699	1.0000	0.9847	161
Apple Granny Smith 1	1.0000	0.9939	0.9969	164
Apple Pink Lady 1	0.9915	1.0000	0.9957	234
Apple Red 1	0.7874	0.9013	0.8405	152
Apple Red 2	0.9747	0.9390	0.9565	164
Apple Red 3	0.9879	0.9939	0.9909	164
Apple Red Delicious 1	0.9156	0.9792	0.9463	144
Apple Red Yellow 1	1.0000	0.9940	0.9970	166
Apple Red Yellow 2	1.0000	0.9939	0.9969	164
Apple hit 1	0.9775	0.9909	0.9841	219
Apricot 1	0.9591	1.0000	0.9791	164
...				
accuracy			0.9619	23619
macro avg	0.9636	0.9614	0.9611	23619
weighted avg	0.9639	0.9619	0.9615	23619

Fara augumentare

	precision	recall	f1-score	support
Apple 6	0.9937	1.0000	0.9968	157
Apple Braeburn 1	0.8862	0.9024	0.8943	164
Apple Crimson Snow 1	0.9773	0.8716	0.9214	148
Apple Golden 1	0.9937	0.9875	0.9906	160
Apple Golden 2	0.9878	0.9878	0.9878	164
Apple Golden 3	1.0000	1.0000	1.0000	161
Apple Granny Smith 1	1.0000	1.0000	1.0000	164
Apple Pink Lady 1	1.0000	1.0000	1.0000	234
Apple Red 1	0.8448	0.9671	0.9018	152
Apple Red 2	0.7816	0.9817	0.8703	164
Apple Red 3	0.9866	0.8963	0.9393	164
Apple Red Delicious 1	0.9724	0.9792	0.9758	144
Apple Red Yellow 1	0.9814	0.9518	0.9664	166
Apple Red Yellow 2	0.9808	0.9329	0.9563	164
Apple hit 1	0.9864	0.9954	0.9909	219
Apricot 1	0.9477	0.9939	0.9702	164
...				
accuracy			0.9603	23619
macro avg	0.9634	0.9588	0.9587	23619
weighted avg	0.9626	0.9603	0.9592	23619

- Precision (With Aug): 0.9639, Without Aug: 0.9626
- Recall (With Aug): 0.9619, Without Aug: 0.9603
- F1 Score (With Aug): 0.9615, Without Aug: 0.9592

Task 4

	precision	recall	f1-score	support
Apple 6	0.7212	0.9554	0.8219	157
Apple Braeburn 1	0.8592	0.7439	0.7974	164
Apple Crimson Snow 1	0.7554	0.9392	0.8373	148
Apple Golden 1	0.9877	1.0000	0.9938	160
Apple Golden 2	0.8410	1.0000	0.9136	164
Apple Golden 3	0.7430	0.9876	0.8480	161
Apple Granny Smith 1	0.9760	0.9939	0.9849	164
Apple Pink Lady 1	0.7312	1.0000	0.8448	234
Apple Red 1	0.6542	0.4605	0.5405	152
Apple Red 2	0.6936	0.9939	0.8170	164
Apple Red 3	0.9310	0.9878	0.9586	164
Apple Red Delicious 1	0.7337	0.9375	0.8232	144
Apple Red Yellow 1	1.0000	0.8735	0.9325	166
Apple Red Yellow 2	1.0000	0.8841	0.9385	164
Apple hit 1	0.9732	0.9954	0.9842	219
Apricot 1	1.0000	0.8963	0.9453	164
Avocado 1	0.9930	0.9860	0.9895	143
...				
accuracy			0.8873	23619
macro avg	0.9026	0.8845	0.8823	23619
weighted avg	0.8987	0.8873	0.8818	23619

Concluzie

Performanța bună a MLP pe imaginile originale și a DeepConvNet (cu și fără augmentare) pe setul de date Fruits-360 se datorează caracteristicilor bine definite ale datelor și adaptării modelelor la sarcina de clasificare. Fruits-360 are imagini curate, bine centrate, de dimensiuni uniforme, fără zgomot vizual și cu clase bine balansate, ceea ce face ca atât modelele simple, cât și cele complexe să poată învăța eficient.

MLP performează bine pentru că imaginile preprocesate permit învățarea directă a corelațiilor dintre pixeli și etichete, iar dimensiunea mică a intrărilor face problema tractabilă. În cazul DeepConvNet, rețeaua este capabilă să extragă caracteristici locale complexe (texturi, margini, forme), ceea ce o face ideală pentru clasificarea imaginilor. Augmentarea îmbunătățește performanța DeepConvNet prin reducerea overfitting-ului și creșterea diversității datelor, ceea ce duce la un model mai robust.

5.2 Fashion-MNIST

SVM

	precision	recall	f1-score	support
0	0.78	0.83	0.81	1000
1	0.97	0.95	0.96	1000
2	0.71	0.72	0.72	1000
3	0.83	0.86	0.85	1000
4	0.72	0.72	0.72	1000
5	0.94	0.92	0.93	1000
6	0.59	0.54	0.57	1000
7	0.89	0.92	0.90	1000
8	0.97	0.96	0.97	1000
9	0.95	0.94	0.94	1000
accuracy			0.84	10000
macro avg	0.84	0.84	0.84	10000
weighted avg	0.84	0.84	0.84	10000

GradientBoosted Trees

	precision	recall	f1-score	support
0	0.80	0.82	0.81	1000
1	0.97	0.94	0.96	1000
2	0.72	0.71	0.72	1000
3	0.83	0.85	0.84	1000
4	0.72	0.72	0.72	1000
5	0.93	0.93	0.93	1000
6	0.57	0.54	0.55	1000
7	0.90	0.90	0.90	1000
8	0.95	0.96	0.96	1000
9	0.94	0.94	0.94	1000
accuracy			0.83	10000
macro avg	0.83	0.83	0.83	10000
weighted avg	0.83	0.83	0.83	10000

Task 1

	precision	recall	f1-score	support
T-shirt/Top	0.7463	0.8120	0.7778	1000
Trouser	0.9335	0.9260	0.9297	1000
Pullover	0.6229	0.6790	0.6498	1000
Dress	0.7812	0.8140	0.7973	1000
Coat	0.6591	0.6710	0.6650	1000
Sandal	0.9135	0.8870	0.9001	1000
Shirt	0.5040	0.3810	0.4339	1000
Sneaker	0.8719	0.8920	0.8819	1000
Bag	0.9334	0.9390	0.9362	1000
Ankle Boot	0.9221	0.9350	0.9285	1000
accuracy			0.7936	10000
macro avg	0.7888	0.7936	0.7900	10000
weighted avg	0.7888	0.7936	0.7900	10000

Task 2

	precision	recall	f1-score	support
T-shirt/Top	0.8377	0.8360	0.8368	1000
Trouser	0.9807	0.9660	0.9733	1000
Pullover	0.7986	0.7970	0.7978	1000
Dress	0.8502	0.8910	0.8701	1000
Coat	0.7854	0.8090	0.7970	1000
Sandal	0.9650	0.9370	0.9508	1000
Shirt	0.7068	0.6630	0.6842	1000
Sneaker	0.9224	0.9510	0.9365	1000
Bag	0.9718	0.9650	0.9684	1000
Ankle Boot	0.9444	0.9520	0.9482	1000
accuracy			0.8767	10000
macro avg	0.8763	0.8767	0.8763	10000
weighted avg	0.8763	0.8767	0.8763	10000

Task 3

Cu augumentare

	precision	recall	f1-score	support
T-shirt/Top	0.8657	0.7480	0.8026	1000
Trouser	0.9849	0.9810	0.9830	1000
Pullover	0.7289	0.9140	0.8110	1000
Dress	0.8429	0.8100	0.8261	1000
Coat	0.5800	0.9280	0.7138	1000
Sandal	0.9953	0.8500	0.9169	1000
Shirt	0.9038	0.3100	0.4617	1000
Sneaker	0.8956	0.9610	0.9272	1000
Bag	0.9876	0.9530	0.9700	1000
Ankle Boot	0.8908	0.9710	0.9292	1000
accuracy			0.8426	10000
macro avg	0.8676	0.8426	0.8341	10000
weighted avg	0.8676	0.8426	0.8341	10000

Fara augumentare

	precision	recall	f1-score	support
T-shirt/Top	0.8424	0.8390	0.8407	1000
Trouser	0.9928	0.9620	0.9771	1000
Pullover	0.8390	0.8440	0.8415	1000
Dress	0.8851	0.8470	0.8656	1000
Coat	0.8058	0.8550	0.8297	1000
Sandal	0.9701	0.9750	0.9726	1000
Shirt	0.6981	0.7170	0.7074	1000
Sneaker	0.9433	0.9480	0.9456	1000
Bag	0.9796	0.9620	0.9707	1000
Ankle Boot	0.9617	0.9540	0.9578	1000
accuracy			0.8903	10000
macro avg	0.8918	0.8903	0.8909	10000
weighted avg	0.8918	0.8903	0.8909	10000

- Precision (With Aug): 0.8676, Without Aug: 0.8918
- Recall (With Aug): 0.8426, Without Aug: 0.8903
- F1 Score (With Aug): 0.8341, Without Aug: 0.8909

Task 4

	precision	recall	f1-score	support
T-shirt/Top	0.7800	0.8050	0.7923	1000
Trouser	0.9928	0.9600	0.9761	1000
Pullover	0.7792	0.7730	0.7761	1000
Dress	0.8518	0.8910	0.8710	1000
Coat	0.7817	0.7880	0.7849	1000
Sandal	0.9439	0.9420	0.9429	1000
Shirt	0.6528	0.6110	0.6312	1000
Sneaker	0.9213	0.9250	0.9232	1000
Bag	0.9474	0.9550	0.9512	1000
Ankle Boot	0.9336	0.9420	0.9378	1000
accuracy			0.8592	10000
macro avg	0.8585	0.8592	0.8587	10000
weighted avg	0.8585	0.8592	0.8587	10000

Concluzie

Setul de date Fashion-MNIST este bine structurat, imaginile fiind centrate și standardizate, ceea ce face ca atât MLP, cât și DeepConvNet să învețe rapid caracteristicile esențiale.

MLP funcționează bine pe imaginile originale datorită dimensiunilor mici și a distribuției uniforme a informației. În plus, datele sunt suficient de curate, permițând modelului să extragă trăsături globale fără a necesita o procesare spațială complexă.

DeepConvNet, mai ales fără augmentare, profită de capacitatea sa de a învăța texturi și forme locale, esențiale pentru diferențierea claselor. Deoarece datele sunt deja bine prelucrate, augmentările pot introduce zgomot inutil, făcând modelele să generalizeze mai bine fără ele.