Proiect – Etapa 2

Deadline: 12.01.2025

Data postării: 9 Decembrie 2024

Contents

1 Introducere

1.1 Scopul proiectului

Continuând eforturile din prima etapă a proiectului, în care ați implementat componentele unui pipeline robust ce permite încărcarea, gestionarea datasetului **Brain Tumor Classification (MRI)** și antrenarea unui model neural, scopul etapei actuale este să optimizăm procesul de antrenare prin aplicarea unor tehnici avansate ce pot fi utilizate pentru optimizarea performanțelor. Clasificarea corectă a imaginilor medicale este crucială în diagnosticul și tratamentul personalizat al pacienților, iar acest proiect își propune să exploreze soluții eficiente și robuste, adaptate specificului setului de date și provocărilor din domeniul medical. Prin utilizarea de strategii precum balansarea claselor, augmentarea imaginilor și aplicarea de modele pre-antrenate, proiectul urmărește să obțină rezultate superioare, relevante pentru implementări practice.

Un aspect esențial al proiectului este realizarea unui **ablation study**, o practică indispensabilă în procesul de antrenare a modelelor de învățare automată. Studiul de

ablație permite analizarea contribuției fiecărei componente a pipeline-ului de antrenare, fie că este vorba despre o metodă specifică de augmentare, o tehnică de regularizare sau o funcție de pierdere. Prin eliminarea sau modificarea treptată a acestor elemente, putem identifica care dintre ele aduc un aport semnificativ la performanța modelului și care sunt mai puțin utile. Această abordare nu doar că optimizează pipeline-ul de antrenare, dar asigură și o înțelegere profundă a relației dintre deciziile de proiectare și rezultatele obținute, un aspect crucial mai ales în aplicațiile sensibile, precum cele din imagistica medicală.

2 Cerințe

2.1 Cerința 1

Pentru a facilita evaluarea robustă și a asigura capacitatea de generalizare a modelului neural propus în cadrul acestui proiect, adaptați pipeline-ul de antrenare pentru a integra validarea de tip ***k-fold cross-validation*** (de exemplu, `k=5`). Acest proces presupune împărțirea setului de date de antrenare în `k` subseturi egale, fiecare subset fiind utilizat pe rând pentru validare, în timp ce celelalte `k-1` subseturi sunt utilizate pentru antrenare.

Pipeline-ul trebuie să includă următoarele componente:

- Implementați o metodă de împărțire a datelor care să distribuie echitabil exemplele între cele `k` fold-uri, păstrând proporțiile claselor din setul de date original.
- Pentru fiecare fold:
  - Antrenați modelul folosind subseturile de date corespunzătoare pentru antrenare și validare.
  - Generați și salvați ploturi care ilustrează evoluția pierderii (*loss*) și a acurateței (*accuracy*) pentru seturile de antrenare și validare, pentru fiecare epocă.
  - Evaluați performanța pe setul de testare (independent de fold-uri) utilizând următoarele metrici: *Precision*, *Recall*, *F1-Score* și *Accuracy*.
- După finalizarea tuturor fold-urilor:
  - Calculați media și deviația standard pentru fiecare metrică (*Precision*, *Recall*, *F1-Score*, *Accuracy*) pe toate fold-urile, atât pentru validare, cât și pentru testare.
  - Documentați și prezentați aceste valori într-un tabel, care să permită o comparație clară între fold-uri și să ofere o perspectivă agregată asupra performanței modelului.

Acest proces va permite o estimare mai precisă a performanței modelului și va reduce influența eventualelor dezechilibre sau particularități din seturile de date. În raportul final, discutați concluziile rezultate și observațiile privind stabilitatea și generalizarea modelului.

Pentru k veți folosi valoarea 5. Setul folosit la antrenare va avea o proporție de 80% și cel pentru validare va avea o proporție de 20%.

2.2 Cerința 2

Pentru a evalua eficiența metodelor propuse de balansare a claselor în cadrul acestui proiect, implementați și comparați performanțele modelului pentru cazurile **cu** și **fără** utilizarea acestor metode. Se vor aplica următoarele tehnici de balansare a claselor:

- **Funcția de pierdere (*loss function*) cu ponderi (weights)**: ajustați funcția de pierdere pentru a penaliza mai mult erorile asociate claselor minoritare.
- **Oversampling**:
  - Cu augmentări aplicate **înainte** de salvarea datelor suplimentare.
  - Fără augmentări suplimentare, folosind doar duplicarea exemplelor din clasele minoritare.

2.3 Cerința 3

Explorați impactul augmentărilor asupra performanței modelului utilizând biblioteca MONAI. Implementați și testați diferite seturi de transformări pentru datele de intrare, având ca obiectiv creșterea performanței modelului. Sarcina presupune:

1. **Definirea augmentărilor:** Pornind de la transformatele analizate etapa trecută, definiți cel puțin trei seturi diferite de transformări. Puteți utiliza transformări din `monai.transforms`.
2. **Antrenarea și evaluarea:** Pentru fiecare set de transformări:
   (a) Antrenați modelul utilizând fold-urile definite anterior (`k-fold validation`).
   (b) Salvați și afișați metricile de performanță pentru fiecare fold, inclusiv:
   - Evoluția funcției (*loss*) pe `train` și `validation`.
   - Acuratețea (*accuracy*) pe `train` și `validation`.
   - Valorile medii pentru metricile *Precision*, *Recall*, *F1-Score* și *AUC* pentru setul de testare.
3. Realizați un tabel comparativ al performanțelor pentru cele trei seturi de augmentări. Indicați în concluzie care set a adus cele mai bune îmbunătățiri și justificați răspunsul.

2.4 Cerința 4

Pentru a îmbunătăți performanța modelului și a preveni overfitting-ul, implementați și testați următoarele tehnici:

1. **Early Stopping:** Implementați mecanismul de *early stopping* pentru a opri antrenarea modelului atunci când performanța pe setul de validare nu mai este îmbunătățită de-a lungul unui anumit număr de epoci (până la 3 epoci consecutive, de exemplu). Aceasta ajută la prevenirea overfitting-ului și la economisirea de resurse.
2. **LR Scheduler:** Adăugați un *learning rate scheduler* pentru a ajusta rata de învățare pe parcursul procesului de antrenare. Exemple de schedulere de rata de învățare ce pot fi utilizate includ:
   - `StepLR` – scade rata de învățare cu un factor la fiecare număr fix de epoci.
   - `ReduceLROnPlateau` – scade rata de învățare când performanța pe setul de validare nu se îmbunătățește.

Pentru fiecare tehnică:

1. Antrenați modelul utilizând fold-ul pentru care ați obținut cele mai proaste performanțe pe setul de validare.
2. Afișați și analizați performanța modelului pentru fiecare metodă implementată, inclusiv pierderea și acuratețea pe seturile de `train` și `validation`.
3. Comparați rezultatele obținute pentru fiecare tehnică și justificați utilizarea lor în contextul problemei.
4. Măsurați timpul necesar procesului de antrenare și raportați-l în tabelul cu rezultatele metricilor de evaluare.

2.5 Cerința 5

După ce ați obținut rezultatele de la experimentele anterioare folosind `k-fold validation`, efectuați un *ablation study* pentru fold-ul care a obținut cele mai slabe rezultate, cu scopul de a analiza impactul diferitelor componente ale modelului asupra performanței. Acest *ablation study* presupune evaluarea modului în care fiecare element (de exemplu, hiperparametri, tehnici de augmentare, regularizare) influențează rezultatele modelului.

Pașii pentru realizarea ablation study sunt următorii:

1. **Selectarea fold-ului cu cele mai slabe rezultate:**

   Identificați fold-ul care a avut cele mai slabe rezultate în termeni de metrici de performanță (precum Acuratețea, Precision, Recall, F1-score).

2. **Experimentare prin eliminarea unor componente:**

   În acest studiu, veți elimina sau modifica anumite componente ale modelului și veți evalua impactul asupra performanței. Exemple de componente ce pot fi testate:

- **Ajustarea funcției de pierdere:** Încercați să utilizați o funcție de pierdere diferită (spre ex. înlocuind `CrossEntropyLoss` cu `BCEWithLogitsLoss`).
- **Focal Loss**: puteți utiliza această funcție de pierdere pentru a acorda prioritate exemplelor dificil de clasificat.

  Focal Loss este o extensie a funcției Cross-Entropy Loss și este definită astfel:

  $$FL(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t)$$

  unde:
  - p_t reprezintă probabilitatea prevăzută corect pentru clasa țintă:

    $$p_t = \begin{cases} p, & \text{dacă eticheta este pozitivă;} \\ 1-p, & \text{dacă eticheta este negativă.} \end{cases}$$

  - α_t este un factor de ponderare pentru clase, utilizat pentru a acorda o importanță mai mare claselor minoritare.
  - $\gamma \geq 0$ este factorul de focalizare care reduce contribuția exemplelor ușor clasificate, concentrând atenția pe cele mai dificile.

  Pentru α și γ, puteți alege valori precum: $\alpha \in \{0.25, 0.5, 0.75\}$, $\gamma \in \{1.0, 2.0, 3.0\}$.
- **Augmentări:** Testați seturi diferite de augmentări ale imaginilor, adică adăugați sau eliminați augmentările existente.
- **Optimizator (Optimizer):** Testați diferiți algoritmi de optimizare, cum ar fi `SGD`, `Adam`, `RMSprop`. Alegeți un optimizator care se potrivește cel mai bine cu tipul de model pe care îl antrenați. Va trebui să alegeți 2 variante de optimizator pe care să le testați.
- **Dimensiunea batch-ului (Batch size):** Experimentați cu dimensiuni diferite ale batch-ului, de exemplu `16`, `32`, `64`, pentru a observa cum influențează performanța și timpul de antrenare.
- **Regularizare:** Testați diferite metode de regularizare pentru a preveni overfitting-ul, inclusiv `Dropout`, `L2 regularization` (sau weight decay), și `early stopping`.

3. **Evaluarea rezultatelor:** După fiecare modificare, evaluați modelul folosind aceleași metrici de performanță, incluzând:
   - **Acuratețea** pe setul de validare,
   - **Precision**, **Recall**, **F1-score** pentru fiecare clasă.
   - **Curba ROC** sau **AUC** pentru a evalua performanța pe fiecare clasă.
4. **Interpretarea rezultatelor:** După realizarea experimentelor de ablație, analizați și interpretați rezultatele pentru a identifica factorii cheie care au contribuit la performanța slabă a fold-ului selectat.

Interpretarea rezultatelor:

În urma *ablation study*, ar trebui să includeți următoarele aspecte în analiza rezultatelor:

1. **Comparația între versiuni ale modelului:** Comparați fiecare experiment de ablație cu modelul de bază (cel care a obținut cele mai slabe rezultate). De exemplu, dacă eliminarea unui tip de augmentare îmbunătățește performanța, acest lucru sugerează că augmentarea respectivă nu aduce beneficii pentru acest set de date sau poate cauza overfitting.

2. **Impactul modificărilor hiperparametrilor:** Dacă modificarea unui hiperparametru, precum rata de învățare, duce la o îmbunătățire semnificativă a performanței, acesta poate indica faptul că modelul are nevoie de o ajustare mai fină a acestuia pentru a învăța mai eficient.

3. **Performanța pe clasele minoritare:** Dacă tehnicile de ponderare a pierderii (de exemplu, `class weights`) au îmbunătățit performanța pe clasele minoritare, aceasta sugerează că modelul a avut dificultăți în a învăța aceste clase din cauza unui dezechilibru în date.

4. **Impactul regularizării și augmentării:** Modificarea tehnicilor de regularizare și augmentare poate arăta dacă modelul suferă de overfitting (de exemplu, când adăugarea `dropout` duce la îmbunătățirea generalizării). De asemenea, analizați ce augmentări au fost cele mai benefice pentru îmbunătățirea performanței.

5. **Comparația metricilor:** Dacă o modificare a determinat o creștere semnificativă a unei metrici precum F1-score, analizați acest aspect pentru a înțelege ce anume a ajutat la îmbunătățirea acesteia (de exemplu, ajustarea funcției de pierdere).

6. **Graficul comparației între fold-uri:** Un grafic comparativ al pierderii și acurateței între fold-ul cu cele mai slabe rezultate și celelalte fold-uri ar putea oferi o imagine clară a modului în care modificările specifice au ajutat la îmbunătățirea performanței pe acest fold.

Va trebui să alegeți 3 aspecte din perspectiva cărora veți realiza acest **ablation study**.

2.6 Bonus 1

În cadrul acestui proiect, pentru a înțelege mai bine cum ia deciziile modelul antrenat, aplicați tehnici de Explainable AI (XAI) pentru a vizualiza regiunile relevante din imagini care au influențat predicțiile modelului. O tehnică utilizată frecvent în acest scop este Class Activation Maps (CAM), care permite vizualizarea zonelor din imagine care au activat cel mai mult unitățile de la nivelul final al rețelei pentru a face o predicție specifică.

Pașii pentru realizarea cerinței:

1. **Aplicarea CAM:**
   - Utilizați tehnica CAM pentru a vizualiza activarea neuronilor relevanți pentru fiecare clasă în parte.
   - Aplicați CAM pentru 5 imagini corect clasificate și 5 imagini incorect clasificate pentru fiecare dintre cele patru clase din setul de date.
   - Vizualizați și salvați hărțile de activare pentru fiecare imagine, astfel încât să puteți observa ce zone din imagini au influențat deciziile modelului. Includeți aceste rezultate în raportul final
2. **Analiza vizualizărilor:**
   - După ce ați generat hărțile CAM, analizați ce regiuni din imagini sunt cele mai influente pentru predicțiile corecte și cele greșite.
   - Verificați dacă modelul se concentrează pe regiuni relevante pentru fiecare clasă sau dacă există distrageri (ex: modelul se poate concentra pe fundalul imaginii în loc de zonele relevante pentru diagnostic).

Pentru implementarea tehnicilor de tip CAM (Class Activation Mapping), puteți porni de la resursele oferite în repository-ul GitHub PyTorch-Grad-CAM (Link). Acest repository include exemple bine documentate pentru diverse metode de vizualizare a activărilor, cum ar fi Grad-CAM, Grad-CAM++ și Score-CAM. Aceste tehnici sunt esențiale pentru interpretarea modelelor de deep learning, oferind o înțelegere vizuală a zonelor din imagine pe care modelul le consideră relevante pentru clasificare. Codul din repository poate fi adaptat cu ușurință pentru setul de date utilizat în proiect, facilitând aplicarea pe exemple corect și incorect clasificate.

2.7 Bonus 2

În această cerință, veți utiliza un model preantrenat (de exemplu, `ResNet18`, `InceptionV3`, sau `EfficientNet`) și veți experimenta cu antrenarea sa pe setul de date specificat, testând performanțele cu diverse configurări ale rețelei. Veți explora două abordări principale: **înghețarea** (frozen) și **dezghețarea** (unfrozen) backbone-urilor rețelei preantrenate, precum și testarea variabilității modelelor.

1. **Alegerea unui model preantrenat:**
   - Alegeți unul dintre următoarele modele preantrenate: `ResNet18`, `InceptionV3`, sau `EfficientNet`.
   - Încărcați modelul preantrenat din PyTorch (`torchvision.models` pentru `ResNet18` și `InceptionV3`, sau din `efficientnet-pytorch` pentru `EfficientNet`).
2. **Testarea cu Backbone Frozen:**

- Înghețați (frozen) toate straturile rețelei, cu excepția ultimelor (fully connected layer).
- Antrenați modelul doar pe ultimele straturi și observați performanțele pe setul de validare.
- Înregistrați metricile relevante (precum loss, acuratețe) și analizați performanțele.

3. **Testarea cu Backbone Unfrozen:**
   - Dezghețați (unfreeze) toate straturile rețelei.
   - Antrenați întregul model, inclusiv backbone-ul.
   - Comparați performanțele cu cele obținute în pasul anterior, observând dacă există îmbunătățiri sau degradări în performanță.

3 Format soluție

La final, va trebui să încărcați o arhivă în format zip care să conțină următoarele:

- **Raport.pdf** este fișierul în care veți include explicațiile, rezultatele, grafice pentru fiecare cerință în parte.
- Unul sau mai multe fișiere cu codul implementat pentru rezolvarea cerințelor (puteți alege dacă lucrați în fișiere `.py` sau în jupyter notebook).

4 Punctaj

Punctajul pentru cerințele propuse este următorul:

Cerința	Punctaj
Cerința 1	10 puncte
Cerința 2	20 puncte
Cerința 3	20 puncte
Cerința 4	20 puncte
Cerința 5	30 puncte
Bonus 1	20 puncte
Bonus 2	20 puncte

Atenție!

- Proiectul este individual!
- **Deadline-ul pentru această etapă este HARD!**
- Pentru a primi punctajul pentru proiect este obligatorie prezentarea lui în săptămâna dedicată.
- Punctajul pentru această etapă este acordat doar dacă acuratețea obținută este minimum 60% (cu alte cuvinte, aveți o soluție îmbunătățită față de etapa anterioară).
- Dacă realizați doar implementare fără a documenta în raport ceea ce vi se cere, nu veți primi punctajul pentru cerință respectivă.
- Proiectele vor fi verificate anti-plagiat, iar cele detectare drept copiate (colegi / internet) nu vor fi punctate.