Învățare Automată

Tema 2 - 2024 - v1.0

1. Descriere generală

Bazat pe cunoștințele acumulate până acum la curs veți explora prin rezolvarea acestei teme câteva aspecte importante din învățarea automată legate de implementarea, antrenarea și evaluarea unei rețele neurale artificiale pentru probleme de clasificare, atât pentru date tabelare cât și pentru date secvențiale.

Sarcinile voastre de lucru vor solicita **utilizarea unor modele** de machine learning bazate pe rețele neurale pentru a rezolva aceste probleme.

2. Descrierea Seturilor de Date

Pentru problema de clasificare veți utiliza același set de date din Tema 1, cel cu informații despre pacienți care trebuie diagnosticați automat (***Patients***).

Pentru problema de clasificare a secvențelor veți folosi setul de date ***PTB Diagnostic ECG***. Acest set de date este compus din 14552 de secvențe cu semnale ale bătăilor inimii ce trebuie clasificate într-una dintre cele 2 categorii: normale și anormale. Semnalele corespund formelor electrocardiogramei (ECG) ale bătăilor inimii pentru cazul normal și pentru cazurile afectate de diferite aritmii și infarct miocardic. Aceste semnale sunt preprocesate și segmentate, fiecare segment corespunzând unei bătăi de inimă. Un segment are 188 de valori, dintre care unele valori pot fi *pad-uri* finale cu valori de 0. Ultima coloana din cele 188 reprezinta eticheta tipului de aritmie (0/1 - normal/anormal).

Scopul este acela de a implementa algoritmi de clasificare a segmentelor de bătăi de inimă.

Notă: Descărcarea setului de date *PTB Diagnostic ECG* se face de la această adresă.

3. Cerințe

3.1. Explorarea Datelor Secvențiale [3p]

Cum în prima temă ați implementat deja o analiză destul de detaliată a setului de date ***Patients***, nu este necesar acest pas pentru primul set de date, ci doar pentru ***PTB Diagnostic ECG***.

Analize recomandate

1. **Analiza echilibrului de clase**

Realizați un grafic al frecvenței de apariție a fiecărei etichete (clase) în setul de date de antrenare / test, folosind **bar plot** / **count plot**.
Pentru realizarea unor astfel de bar plots puteți folosi mai multe biblioteci:

- Folosind biblioteca seaborn pentru barplot sau countplot
- Direct dintr-un DataFrame Pandas folosind pandas.DataFrame.plot.bar

2. Vizualizarea seriilor de timp

1) **Afișați câte un exemplu de serie** pentru fiecare **categorie de aritmie** din setul de date PTB.

2) Pentru setul de date cu aritmii afișați un grafic al **mediei** și **deviației standard per unitate de timp**, pentru fiecare **clasă de aritmie**. Media și deviația standard se calculează peste toate exemplele (atât din train, cât și din train set).

Notă: Diagramele din această secțiune sunt cele ***minimal cerute*: NU** sunt singurele pe care le puteți face :-)

3.2. Utilizarea modelelor de Rețele Neurale [7p]

Sunt propuse spre evaluare următoarele arhitecturi de rețele neurale:

- **Arhitectură de tip MLP** (Multi-Layered Perceptron)
  - Pentru setul de date ***Patients***, rețeaua primește ca intrare toate datele pacientului **[2p]**
  - Pentru setul de date ***PTB Diagnostic ECG*** primește ca intrare întreaga secvență a unei bătăi de inimă **[2p]**
  - Experimentați cu **numărul de straturi** și **dimensiunea acestora**.

- **Arhitectură de tip convoluțională**, folosind straturi de **convoluții 1D** și **global average pooling**, folosită doar pentru setul de date ***PTB Diagnostic ECG*** **[3p]**
  - Puteți genera propria voastră arhitectură explorând:
    - Numărul de straturi de convoluție și dimensiunea canalelor (channels) a fiecăruia
    - Combinarea cu straturi liniare pentru partea finală a rețelei
    - Exemple de tutoriale găsiți aici și aici.
  - Puteți folosi o arhitectură dată, **InceptionTime**, care utilizează straturi convoluționale 1D adaptând arhitectura Inception (definită pentru imagini) pe cazul seriilor de timp.
    - Implementare în Pytorch pentru InceptionTime
    - Implementare în Keras pentru InceptionTme
    - Sugestii:
      - Folosiți 1 modul de Inception (în loc de default-ul de 2 din paper)
      - Variați dimensiunea kernelelor de convoluție

NOTĂ: Datele pe care le aveți la antrenare sunt relativ puține din punct de vedere numeric. Nu uitați că trebuie să le împărțiți în seturi de antrenare și testare (sau antrenare / validare / testare). **Luați în considerare** (a se observa și în tutorialele referențiate) utilizarea metodelor de regularizare, e.g. prin utilizarea straturilor de **Dropout** sau prin utilizarea unui mecanism de weight decay în optimizator (a se vedea detalii aici).

NOTĂ: În afară de arhitectura în sine, performanța unui model neural este influențată și de **optimizatorul ales și de parametrizarea acestuia**. **Sugestii**:
- Folosiți un optimizator adaptiv (e.g. ADAM) și unul cu rată de învățare (learning rate) configurabilă (e.g. SGD cu momentum și un Learning Rate Scheduler)
- Explorați influența **dimensiunii batch-urilor**
- Explorați influența numărului de epoci de antrenare

Evaluarea Modelelor

În raportul vostru trebuie să includeți următoarele:

- Pentru fiecare model trebuie inclusă o detaliere a setup-ului de antrenare:
  - Descrierea arhitecturii folosite
  - Descrierea **parametrilor de optimizare** (optimizator folosit, batch size, learning rate, learning rate scheduler, weight decay)
- Afișați **pe același grafic** curbele de loss pentru **antrenare și test**
  - După ce ați evaluat mai multe *variațiuni* ale aceleiași arhitecturi generale (e.g. un model conv1D cu dimensiune diferită a kernelelor de convoluție), trasați curbele de loss la antrenare și test pentru fiecare variație **pe același grafic**, astfel încât să se poată observa diferențele între ele
- Creați un **tabel** în care să indexați **pe linii** configurațiile arhitecturale și de optimizare încercate, iar pe coloane metricile de performanță (acuratețea generală de clasificare, precizie / recall / F1 la nivelul fiecărei clase în parte)
- Creați matricea de confuzie pentru clasificarea celor 5 tipuri de aritmii (MIT-BIH) și pentru clasificarea între normal și anormal a datelor din PTB.

4. Predarea temei

Tema va fi încărcată pe Moodle însoțită de un raport sub formă de fișier PDF, care include:

- **Cerința 3.1** - cuprinde toate vizualizările și statisticile cerute. **Este obligatorie** prezența în text a **unei interpretări / analize** a diagramelor rezultate.
- **Cerința 3.2** - include raportarea evaluării rețelelor neurale de clasificare pentru cele două Seturi de date propuse (***Patients*** și ***PTB Diagnostic ECG***). **Este obligatorie** prezența în text a **unei interpretări / analize** a rezultatelor obținute (e.g. care este influența arhitecturii, cât de puternic este impactul hiper-parametrilor asupra performanței, care sunt clasele cu cele mai bune predicții).

Rezultatele temei vor fi prezentate în cadrul laboratoarelor de Învățare Automată, **exclusiv pe baza rapoartelor încărcate.**