

Documentatie proiect Kaggle

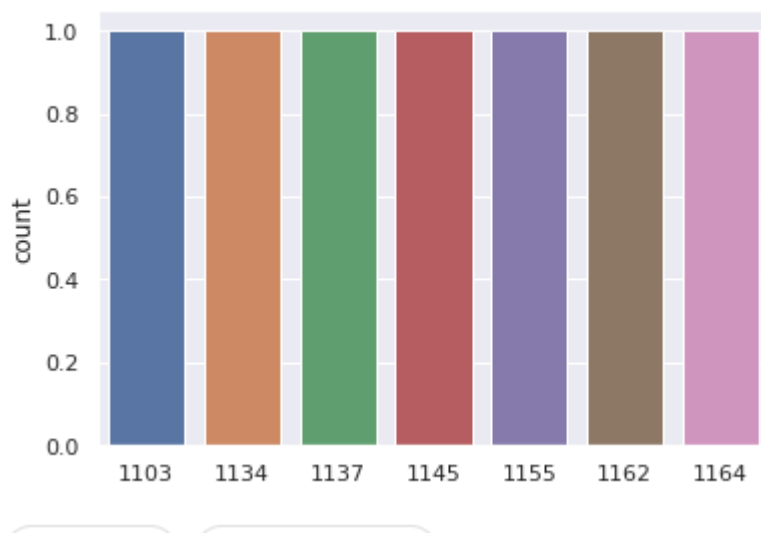
Butelca Radu-Andrei

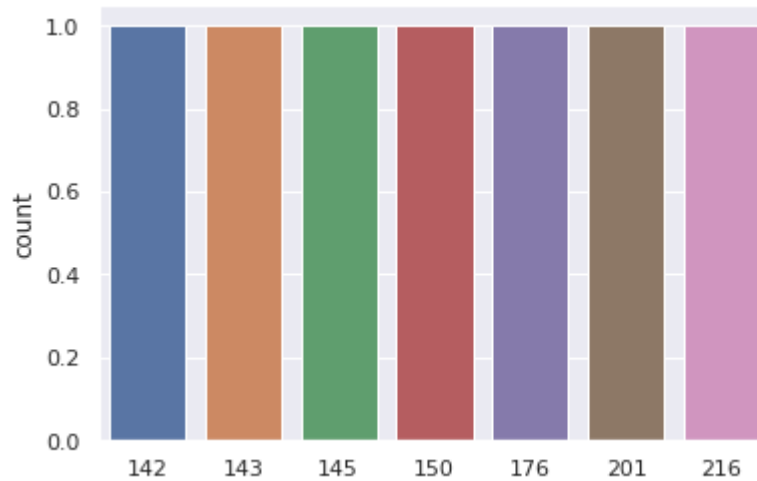
Aprilie 2022

1 Introducere

In cadrul proiectului Kaggle "Deep Hallucination Classification" am avut de clasificat imagini cu o rezolutie mica (16, 16, 3) in 7 clase diferite.

La o prima vizualizare a datelor de antrenament am observat ca imaginile nu pot fi clasificate, in mare parte, de ochiul uman, astfel incat este nevoie de un program de inteligenta artificiala pentru o acuratete mai mare a clasificarii. Atat imaginile de validare, cat si cele de antrenament au fost impartite in mod proportional, fiecare clasa avand un numar aproximativ egal de elemente, dupa cum se poate observa in figurile 1 si 2, lucru ce ne fereste de anumite discrepante in vederea datelor. In urma vizualizarii acestor caracteristici, am decis ca cele mai bune modele de Machine Learning pe care le pot utiliza sunt SVM(Support vector machines) si CNN (convolutional neural network).





2 SVM(Support vector machine)

Maschinele vector suport sunt modele de invatare automata supervizata utilizate, in general, in contextul clasificarilor. Dupa ce am importat imaginile din fisierele date si le-am transformat in matrici de pixeli am vazut ca valorile sunt cuprinse intre 0 si 1, ceea ce facea ca **standardizarea** sa nu isi mai aibe rostul, aceasta actiune doar afectand datele, facandu-le supraunitare(prin impartirea la medie ce era un numar subunitar).

Pentru implementarea modelului am utilizat biblioteca **ScikitLearn**, ce vine cu metoda de tip **one vs one**, adica clasele vor fi antrenate doua cate doua rezultand un numar de antrenamente egal cu cel al numarului de muchii dintr-un graf complet: $\frac{n(n-1)}{2}$, n fiind numarul de clase.

Dupa antrenament, atunci cand facem predictiile, eticheta ce este atribuita fiecarei imagini din setul de testare este cea care a obtinut cea mai mare valoare.

2.1 Alegerea parametrilor

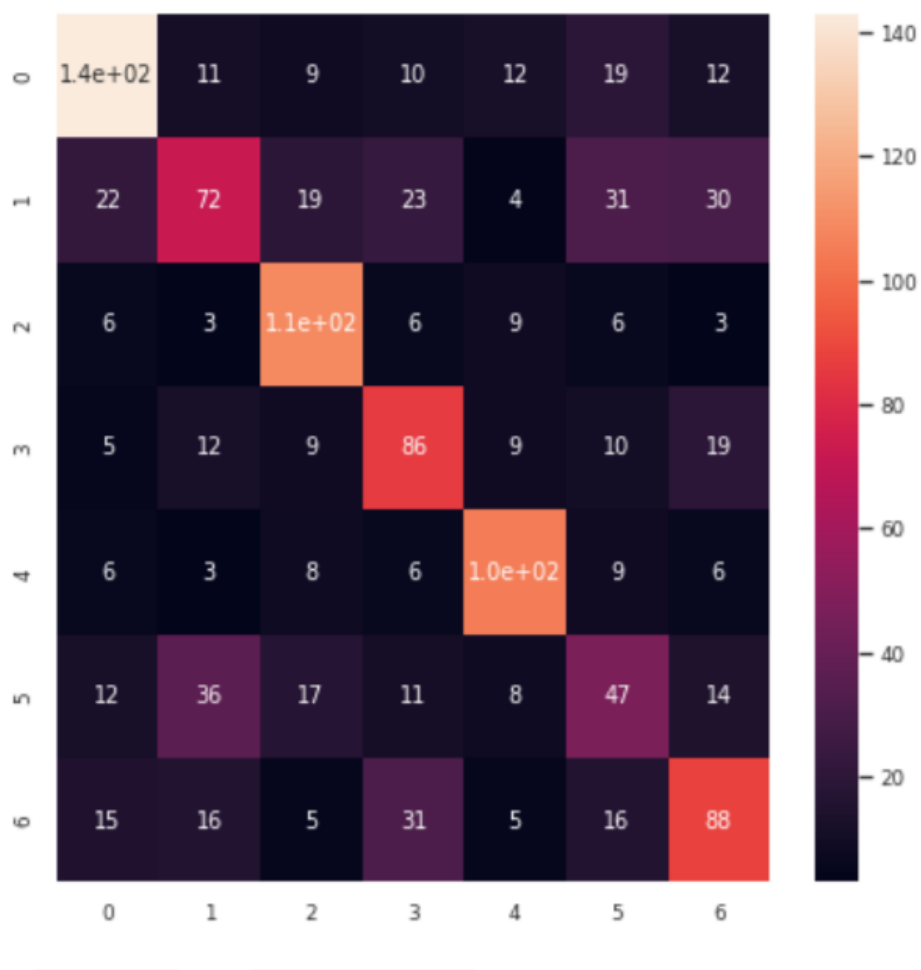
In pofida faptului ca biblioteca importata usureaza munca, a fost nevoie si de ajustarea unor parametri.

In vederea functiei kernel, cea care cauta clasificatori liniari pentru probleme neliniare, am incercat Polynomial Kernel, Gaussian Kernel, Linear Kernel(cel default din model) si Radial Basis Function(rbf), cele mai bune rezultate venind de la cel din urma, maximul acuratetii oferit de celelalte fiind in vecinatatea procentului de 50%.

In cautarea celui mai bun rezultat am ajustat si valorile C si γ printr-o functie ce cauta prin multimea $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$ valorile optime.

| | C | 100 | 10 | 1 | 10^{-1} | 10^{-2} | 10^{-3} |
|-----------|---|-------|-------|-------|-----------|-----------|-----------|
| Gamma | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 100 | 0 | 0.132 | 0.144 | 0.157 | 0.456 | 0.537 | 0.549 |
| 10 | 0 | 0.132 | 0.144 | 0.157 | 0.456 | 0.554 | 0.525 |
| 1 | 0 | 0.130 | 0.143 | 0.156 | 0.446 | 0.542 | 0.468 |
| 10^{-1} | 0 | 0.128 | 0.150 | 0.169 | 0.462 | 0.399 | 0.121 |
| 10^{-2} | 0 | 0.143 | 0.156 | 0.374 | 0.203 | 0.121 | 0.121 |
| 10^{-3} | 0 | 0.121 | 0.121 | 0.121 | 0.121 | 0.121 | 0.121 |

Dupa cum putem observa, cel mai bun rezultat l-am obtinut cu functia kernel "rbf" si pentru $\gamma = 10^{-2}$, $C = 10$, predictia finala avand urmatoarea matrice de confuzie:



3 CNN(Convolutional Neural Network)

Cand vine vorba de clasificarea imaginilor, cel mai buna metoda descoperita pana in prezent este cea care foloseste **rețele neuronale convolutive**, acestea fiind o subclasa a rețelelor neuronale, adaosul pe care il aduc fiind acela al operatiilor de convolutie, pooling, data augmentation, etc. . .

Operatia de **convolutie** este una in care 2 functii produc o a 3-a prin modificarea formei a uneia dintre ele, aducand astfel in prim-plan anumite trasaturi ce contin o doza de interes mai mare.

Operatia de **pooling** este una in care se restrange spatiul de lucru pentru a utiliza memoria mai eficient. Pentru acest proiect am folosit un pooling de tip 'max', ce returneaza pixelul cu cea mai mare valoare pe un interval dat.

Operatia de **data augmentation** este una in care marim setul de date de antrenament prin aplicarea anumitor ajustari asupra imaginilor (zoom, rotatii, etc..) cu scopul de a evita overfitting-ul.

Pana sa folosesc un model de date preantrenat, acuratetea maxima pe care am obtinut-o a fost de 63% indiferent de ce parametri sau optimizatori am folosit. Asadar, am utilizat Xception, deoarece l-am considerat ca a fi cel mai potrivit pentru situatia actuala, timpul de rulare fiind de 8ms pe GPU, dimensiunea de 88 mb, 22.9 milioane de parametri, 79% top-1, 94.5% top-5, cu o adancime de 81 de layere.

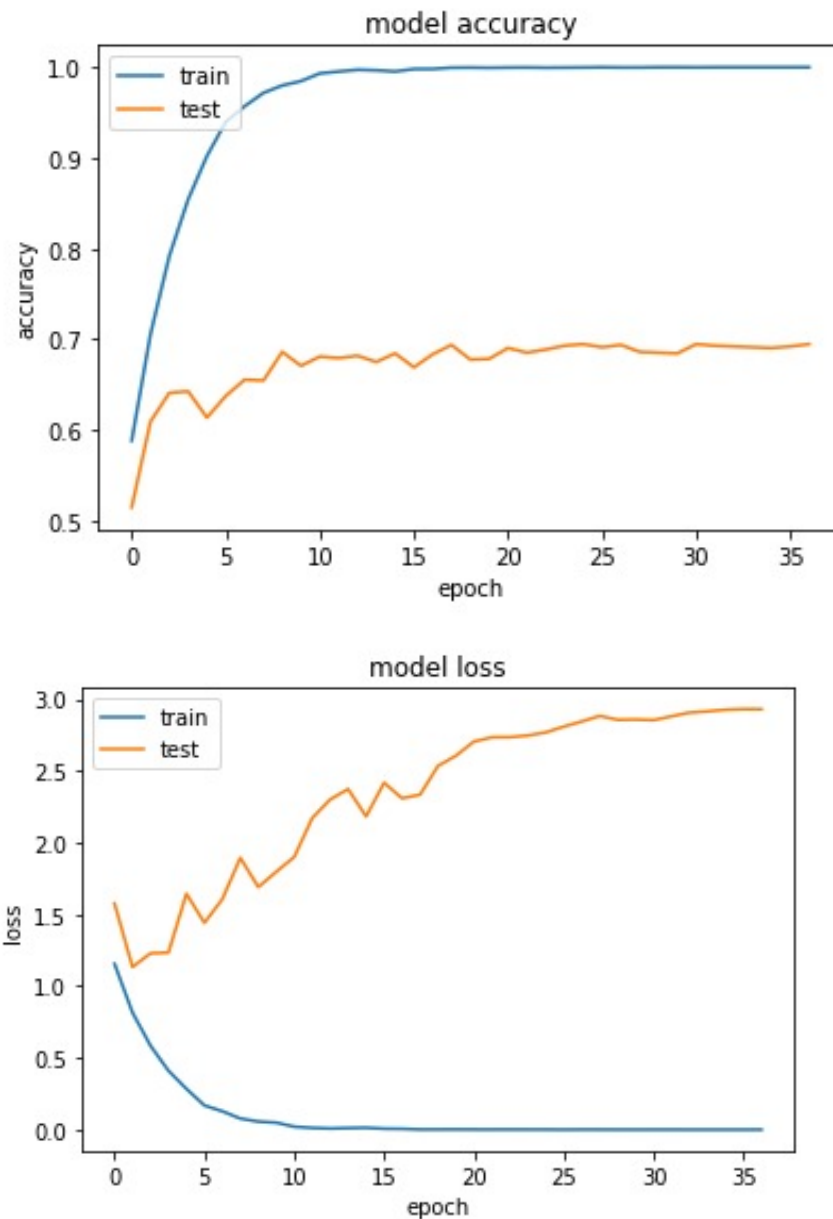
Dupa ce am modificat dimensiunea imaginilor pentru a se putea fi antrenate in Xception si aplicat modelul, am transformat datele in "neuroni". Pe urma, am mai adaugat 3 layere, ultimul fiind de 7, egal cu numarul de clase.

Functiile de activare pe care le-am folosit au fost ReLU, $f(x) = \max(x, 0)$ si softmax, functie de probabilitate ce scaleaza valorile.

Ca si optimizator am folosit Adam cu o rata de invatare de 10^{-4} , deoarece acesta ajunge cel mai rapid la valori de acuratete optima, de aici rezultand dezavantajul pe care l-am avut cu SGD, acesta din urma avand nevoie de mai multe epoci, fapt ce ar fi adus o dificultate in plus, tinand cont de importarea celor mai mult de 14 milioane de imagini din ImageNet.

Din cauza lipsei de memorie, am fost constrans sa utilizez la batch size o valoare de cel mult 64, aceasta aducandu-mi si cele mai multe beneficii.

In urmatoarele imagini putem observa cum s-au desfasurat de-alungul mod-elarii principalii parametri:



S-a putut observa ca dupa aproximativ 10 epoci valoarea loss-ului pentru datele de test a inceput sa creasca, dar si acuratetea pe modelul de antrenament incepuse sa stagneze, ceea ce inseamna ca epoca 10 era una optima pentru algoritmul creat.

Contrar asteptarilor, augmentarea datelor a venit doar cu probleme, prin adaugarea acestora reusind sa scad din procentajul predictiilor.

4 Concluzii

In final, pe modelul SVM am obtinut un scor de 0.54%, in timp ce pe CNN am obtinut 0.70170%, cu mentiunea ca exista, cu siguranta, metode mai bune de a rezolva problema **Deep Hallucination Classification**, una dintre acestea putand fi prin a alege mai bine parametri in cadrul augmentarii datelor.