

Prepoznavanje saobraćajnih znakova i njihova klasifikacija

Uroš Aksentijević
aksaa002@gmail.com

Sažetak. Današnje vreme nemoguće je zamisliti bez tehnologije koju ćemo neprestano razvijati do trenutka kada će ona početi sama da se razvija. Primena mašinskog učenja proširuje naše poglede na polju tehnoglogije, medicine, filologije i ostalih disciplina. Prepoznavanje saobraćajnih znakova i njihova klasifikacija je veoma mali segment mašinskog učenja i veliku primenu ima u sistemu autonomnih vozila. Sistem se zasniva na korišćenjem neuronskih mreža, kao i posebnu vrstu istih za detekciju objekata sa slike.

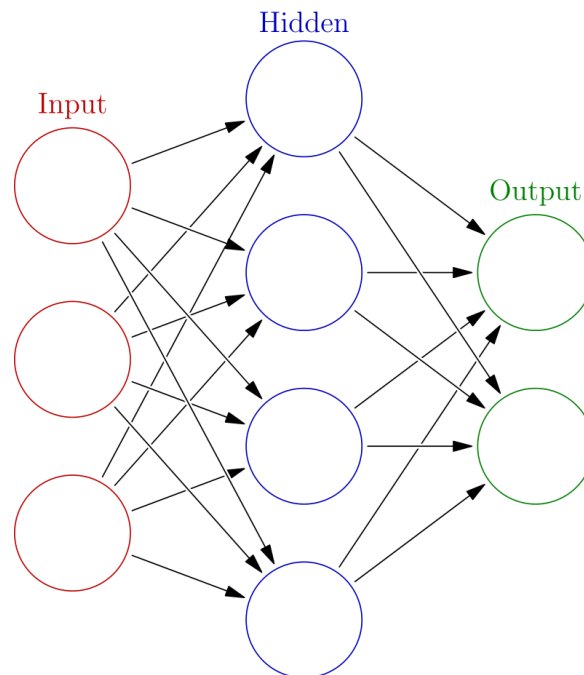
Sadržaj:

Sažetak	2
Sadržaj	3
Uvod u mašinsko učenje	4
Treniranje neuronske mreže	5
Kreiranje baze podataka	7
Treniranje R-CNN	8
Reference	9

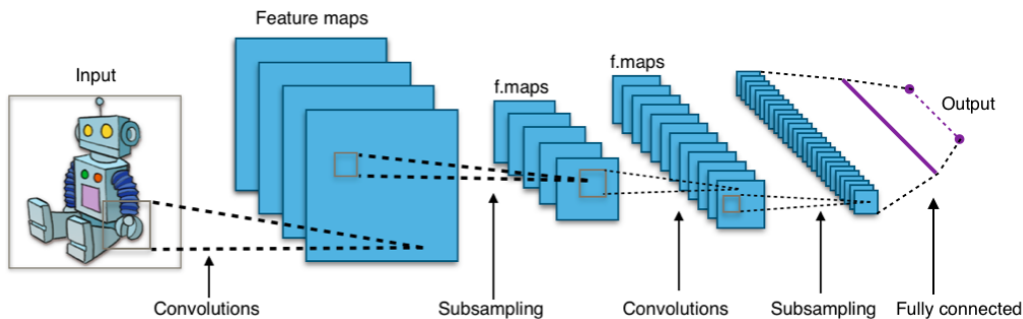
1. Uvod u mašinsko učenje

Mašinsko učenje (*machine learning*; ML) je deo veštačke inteligencije i predstavlja naučnu disciplinu o algoritmima i statističkim modelima koje računarski sistem koristi kako bi izvršio operacije bez eksplicitnih instrukcija oslanjajući se na obrasce i zaključke. Algoritmi mašinskog učenja prave matematičke modele zasnovane na podacima, koji se često nazivaju na engleskom “training data”, kako bi napravili predikcije i odluke bez pravljenja posebnog programa da izvrši neku operaciju.

Veštačka neuronska mreža (en. *Artificial neural networks*; ANN) je model mašinskog učenja i ovaj sistem uči da radi određene operacije bez dodatnog programiranja istog. Sličan je biološkim neuronskim mrežama. ANN zasniva se na kolekciji ujedinjenih čvorova (en. *artificial neurons*), tj. pojednostavljene verzije bioloških neurona u mozgu. Svaka veza između ovih čvorova može da transmutuje signal od jednog do drugog. Neuron koji primi signal može da ga obradi i zatim preda njegov ishod drugim neuronima sa kojima je povezan. Glavni cilj ANN jeste da rešava probleme kao što i ljudski mozak to radi. Primeri ANN-a su: computer vision, speech recognition, filtriranje socijalnih mreža, igranje igrica, medicinske dijagnoze.



Duboko učenje (en. *Deep learning*; DP) je deo mašinskog učenja koji koristi model ANN, tačnije konvolucionarnu neuronsku mrežu (en. *Convolutional neural network*; CNN ili ConvNet) i ona se koristi najviše za analiziranje slika. Takođe koristi se za klasifikaciju slika, analizu medicinskih slika, obradu prirodnog jezika... CNN su izvedene kao višeslojni preceptor (en. *multilayer perceptron*). To znači da je svaki neuron u jednom sloju povezan sa svim neuronima u sledećem sloju (en. *fully connected network*). DP takođe koristi više slojeva kako bi dobio više podataka sa čistog ulaza. Primer, u analizi slike, manji slojevi mogu da identifikuju ivice, dok viši slojevi mogu da prepoznaju cifre, slova ili lica ljudi.



2. Treniranje neuronske mreže

Da bi napravili neuronsku mrežu, potrebno je da imamo veliku bazu podataka (u ovom slučaju bazu slika saobraćajnih znakova). Moramo voditi računa da sve slike moraju biti istih dimenzija i da budu RGB.

Funkcija `imageDatastore`, obezbeđuje nam skladištenje svih slika iz naše baze slika.

```
imds = imageDatastore(filename, 'IncludeSubfolders', true, 'LabelSource', 'foldernames');
```

Sledeći korak je da iz baze koja je sada učitana u promenljivu 'imds', izdvojimo 70% nasumično izabranih slika koje će nam služiti da se na njima naša neuronska mreža uči a preostalih 30% će nam koristiti za validaciju neuronske mreže.

```
[imdsTrain, imdsValidation] = splitEachLabel(imds, 0.7, 'randomize');
```

Sada imamo dve nove baze slika od kojih jedna služi za treniranje mreže (imdsTrain) a druga za njenu validaciju (imdsValidation).

Nakon toga, potrebno je definisati slojeve kroz koje će prolaziti podaci i kroz koje će se obrađivati isti. Sve slike moraju biti istih dimenzija (32x32 RGB). Postoje 43 znaka i njih ćemo međusobno povezati.

```
layers = [  
    imageInputLayer([32 32 3])  
  
    convolution2dLayer(3,8,'Padding','same')  
    batchNormalizationLayer  
    reluLayer  
  
    maxPooling2dLayer(2,'Stride',2)  
  
    convolution2dLayer(3,16,'Padding','same')  
    batchNormalizationLayer  
    reluLayer  
  
    maxPooling2dLayer(2,'Stride',2)  
  
    convolution2dLayer(3,32,'Padding','same')  
    batchNormalizationLayer  
    reluLayer  
  
    fullyConnectedLayer(43)  
    softmaxLayer  
    classificationLayer  
];
```

Sledeće što treba uraditi, jeste postaviti opcije za treniranje naše neuronske mreže. Kako bi tačnost u klasifikaciji znakova bila što veća potrebno je povećati broj epoha i omogućiti što više vremena za treniranje neuronske mreže.

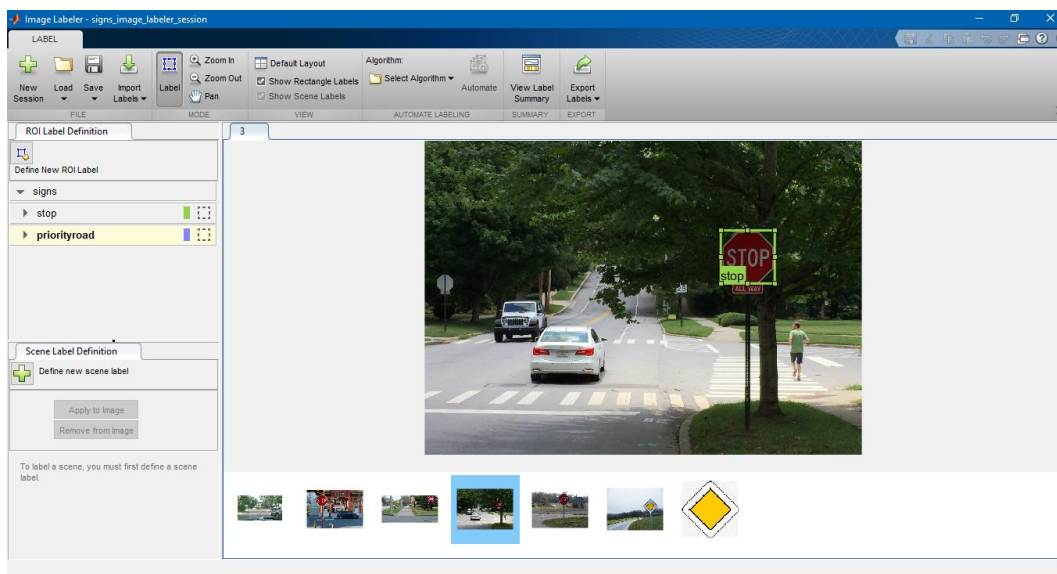
```
options = trainingOptions('sgdm', ...  
    'InitialLearnRate', 0.01, ...  
    'MaxEpochs', 50, ...  
    'Shuffle', 'every-epoch', ...  
    'ValidationData', imdsValidation, ...  
    'ValidationFrequency', 50, ...  
    'Verbose', false, ...  
    'Plots', 'training-progress');
```

Sve što je bilo potrebno oko podešavanja treniranja neuronske mreže je završeno i možemo otpočeti njeno treniranje.

```
net = trainNetwork(imdsTrain, layers, options);
```

3. Kreiranje baze podataka

Sada kada smo istrenirali našu neuronsku mrežu, potrebno je stvoriti novu bazu podataka (bazu slika), koristeći Image Labeler (IL) u Matlabu. IL predstavlja aplikaciju koja služi za formiranje baze podataka (ground truth ili tabele) u kojoj se nalaze podaci o slici i mestima na kojima se nalaze obeleženi objekti u ovom slučaju saobraćajni znakovi. Potrebno je naći veliki broj slika saobraćajnih znakova i pritom ručno obeležiti sve saobraćajne slike koje se nalaze na tim slikama. Postoje i algoritmi koji nakon nekoliko ručno urađenih primera mogu da nastave obeležavanje ostalih saobraćajnih znakova na ostalim slikama. Nakon toga, potrebno je sačuvati bazu kao tabelu pod nazivom 'signs_table' koju ćemo koristiti za detekciju novih znakova sa bilo koje slike.



4. Treniranje R-CNN

Prethodno u podglavlju 1, saznali ste šta su to CNN i sada kada imamo sve podatke potrebne za treniranje ove mreže, možemo isto otpočeti.

Da bi otpočeli treniranje potrebna nam je tabela koju smo kreirali u Image Labeleru i neuronska mreža koja je naučena da prepoznaje saobraćajne znakove sa slika dimenzija 32x32.

Prvo je potrebno postaviti neke od opcija kao kada smo i trenirali neuronsku mrežu za klasifikaciju.

```
options = trainingOptions('sgdm', ...  
    'MiniBatchSize', 128, ...  
    'InitialLearnRate', 1e-3, ...  
    'LearnRateSchedule', 'piecewise', ...  
    'LearnRateDropFactor', 0.1, ...  
    'LearnRateDropPeriod', 100, ...  
    'MaxEpochs', 100, ...  
    'Verbose', true);
```

Nakon toga, možemo otpočeti treniranje R-CNN.

```
rcnn = trainRCNNObjectDetector(signs_table, net, options, ...  
    'NegativeOverlapRange', [0 0.3], 'PositiveOverlapRange',[0.5 1])
```

Ovaj proces će potrajati nekoliko minuta i nakon toga smo završili treniranje R-CNN koja je spremna za detekciju, obeležavanje i klasifikaciju saobraćajnih znakova.

Reference

- [1] Machine learning,
https://en.wikipedia.org/wiki/Machine_learning
- [2] Deep learning,
https://en.wikipedia.org/wiki/Deep_learning
- [3] Artificial neural networks,
https://en.wikipedia.org/wiki/Artificial_neural_network
- [4] Mathworks,
<https://www.mathworks.com/>
- [5] GTSRB, databaza za treniranje neuronske mreže
<http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset>