

Проект: Разпознаване на пътни знаци

1. Въведение

Проектът представлява конволюционна дълбока невронна мрежа за разпознаване и класифициране на пътни знаци в 43 различни категории. Имплементирана е с помощта на TensorFlow, Keras библиотеките.

2. Проблематика

Разпознаването на пътни знаци е критичен компонент от множество системи за умен контрол на пътния трафик, както и в областта на автономното управление на автомобили. Целта е разработката на модел способен с висока точност да разпознава и класифицира различни видове знаци, независимо от външни фактори като променливи нива на осветяването и разнообразни атмосферни условия.

3. Данни

Обучението на модела се извършва върху база със свободен достъп с 26 640 снимки на различни пътни знаци, групирани в 43 различни категории. За всеки вид съществува множество от изображения с различно качество и резолюция, заснети в различни атмосферни условия, под различен ъгъл, на различни места и с варираща осветеност/време от деня.



4. Архитектура на модела

Моделът представлява дълбока конволюционна невронна мрежа, изградена с TensorFlow, Keras.

Съдържа два конволюционни слоя с по 32 филтъра всеки и Kernel 3x3.

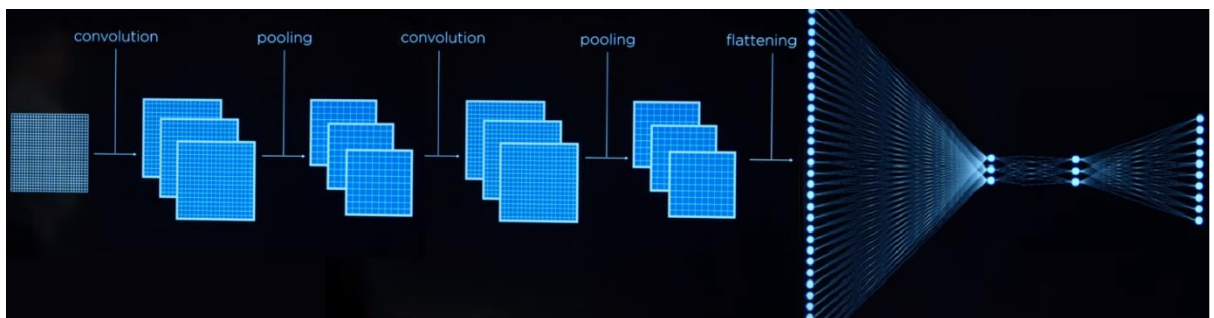
След всеки от двата слоя се извършва MaxPooling с Pool 2x2 с цел смачкване на тренировъчните изображения за по-лесно обучение.

След конволюционния етап данните до момента се представят като едноизмерен масив чрез операцията Flattening. Това действие създава входният слой на мрежата.

Следват два скрити слоя, първият с 172 неврона, а вторият с 86.

Изходният слой се състои от брой неврони равен на броя на категориите, в които можем да класифицираме подаденото изображение.

За да избегнем Overfitting и прекалена зависимост от конкретни неврони прилагаме Dropout техника, премахвайки случайни 30% от невроните в мрежата при всяка тренировъчна итерация.



5. Процес на разработка

- 1) Започнах процеса на разработка на модел за разпознаване и категоризиране на пътни знаци, използвайки предходен свой проект за разпознаване на ръкописни цифри с модификация броя на параметрите в изходния слой, защото фундаментално двата проблема са от едно и също естество.
 - 1 Конволюционен слой с 32 филтъра, 3x3 Kernel
 - 1 Max Pooling с 2x2 Pool
 - 1 Скрит слой с 128 параметъра
 - Изходен слой с 43 параметъра
 - **Точност: 5%**
- 2) Имайки предвид сравнително високата комплексност на проблема реших да увелича броя неврони в скрития слой на 240, за да повиша способността на модела да решава комплексни проблеми и да разреши настъпилия Underfitting .
 - 1 Конволюционен слой с 32 филтъра, 3x3 Kernel
 - 1 Max Pooling с 2x2 Pool
 - 1 Скрит слой с 240 параметъра
 - Изходен слой с 43 параметъра
 - **Точност: 50% на training set-a и 78% на testing set-a**
- 3) Виждайки значителното повишаване в точността на модела прецених да добавя още комплексност под формата на втори скрит слой със 120

параметъра, правейки мрежата дълбока. Това би трябвало да спомага за йерархичното разбиване на данните, което би трябвало да е особено полезно при обработка на изображения, съставени от множество елементи, като например форма и символ върху пътният знак и т.н.

- 1 Конволюционен слой с 32 филтъра, 3x3 Kernel
- 1 Max Pooling с 2x2 Pool
- 1 Скрит слой с 240 параметъра
- 1 Скрит слой с 120 параметъра
- Изходен слой с 43 параметъра
- **Точност: 90-93%**

- 4) Проведах експеримент с променяне на броя филтри в конволюционния слой от 32 на 52, но това не доведе до промени в точността, за това се върнах към 32 филтъра.
- 5) Достигнах до наблюдението, че при всяко ново обучение на модела точността може да варира значително, което понякога налага тренирането на модела да бъде повторено няколко пъти до достигане на оптимално висока точност. След кратко проучване установих, че това е страничен ефект на Dropout техниката, поради, което намалих процента на изключените параметри от 50% на 30%. Това разреши проблема без да създава нови нежелани ефекти.
- 6) Към този момент точността беше задоволителна, но реших да пробвам да олекотя модела, намалявайки броя на параметрите в двата скрити слоя на 172 и 86. Това доведе до спад в точността от 93% на 89%, което сметнах за приемливо.
 - 1 Конволюционен слой с 32 филтъра, 3x3 Kernel
 - 1 Max Pooling с 2x2 Pool
 - 1 Скрит слой с 172 параметъра
 - 1 Скрит слой с 86 параметъра
 - Изходен слой с 43 параметъра
 - **Точност: 89%**
- 7) Спомних си, че конволюция и Pooling могат да се използват повече от един път и прецених да изпробвам използването им втори път в идентичен вид с първата им употреба. Това доведе до подобрение и нова точност от 92-96%.
 - 1 Конволюционен слой с 32 филтъра, 3x3 Kernel
 - 1 Max Pooling с 2x2 Pool
 - 1 Конволюционен слой с 32 филтъра, 3x3 Kernel
 - 1 Max Pooling с 2x2 Pool
 - 1 Скрит слой с 172 параметъра
 - 1 Скрит слой с 86 параметъра
 - Изходен слой с 43 параметъра
 - **Точност: 92-96%**

8) Избрах това състояние на модела за финална версия.