

Klasifikácia vtáčích druhov cez Konvolučnú neurónovú sieť

www.kaggle.com

Obsah

Dátová množina vtáčích druhov	3
Problém klasifikácie vtáčích druhov	4
Návrh architektúry konvolučnej neurónovej siete	6
Experiment č. 1 - Softmax regresia - Klasifikácia druhov vtákov	7
Experiment č. 2 - Softmax regresia - Klasifikácia druhov vtákov	11
Poznámky k implementácii	12

Dátová množina vtáčích druhov

Obrázky na úlohu klasifikácie sú prebraté z databázy Kaggle dostupnej na webovom odkaze [BIRDS 400 - SPECIES IMAGE CLASSIFICATION | Kaggle](https://www.kaggle.com/ann/softmax-cnn) a obsahujú 60 388 RGB (3-kanálových) obrázkov. Celý dataset obsahuje obrázky 400 druhov vtákov. Obrázky vtákov sa nachádzajú vo formáte JPG a obrázky pre každý druh sú organizované v samostatnom priečinku. Celkovo pozostáva z:

- 58 388 obrázkov v trénovacej množine - 79 druhov vtákov.
- 2000 obrázkov vo validačnej množine - 400 druhov vtákov.
- 2000 obrázkov v testovacej množine (5 JPG obrázkov pre každý druh) - 400 druhov vtákov.

Dáta získané z verejnej databázy sú už pri ich získaní organizované do trénovacej, validačnej a testovacej množiny. Na každom obrázku je len 1 vták. Z výsledkov rozdelenia dát autormi vyplýva že sa rozhodli takmer 97 % obrázkov z celej množiny obrázkov použiť na trénovanie modelu a len 3,31 % dostupných obrázkov použiť na validáciu a testovanie modelu. Takáto situácia je však výnimkou ako pravidlom, pretože rozdelenie dát v tomto (alebo obdobnom pomere 98 %, 1 % a 1%) je skôr bežné pre množiny dát s miliónmi vzoriek.

Obrázky sú v RGB troj-kanálovom farebnom modeli s výškou 224 pixelov a šírkou 224 pixelov. Vo všetkých obrázkoch vták o ktorého klasifikáciu máme záujem zaberá 50 % priestoru obrázku. Ten istý druh vtáku a jeho obrázok z trénovacej množiny je tiež možné nájsť vo validačnej a testovacej množine. Z toho vyplýva:

- že na obrázkoch vtákov ktoré sa nachádzajú aj v trénovacej aj vo validačnej množine bude klasifikujúca neurónová sieť pravdepodobne úspešnejšia ako by mala byť.
- na ostatných obrázkoch ktoré neurónová sieť pri trénovaní nevidela je zvýšená pravdepodobnosť že bude pri klasifikácii týchto obrázkov z validačnej množiny neúspešná.

Vzhľadom na to že v trénovacej množine je prevaha vtákov samcov ako samičiek (85 % samcov) a že samce vtákov sú farebnejšie ako samičky jedná sa o množinu dát z nevyváženým množstvom tried do ktorých sa neskôr posnažíme obrázky klasifikovať¹. Bežnou technikou pri obmedzenej dostupnosti záznamov z ďalších tried aplikovanou na množinou dát je vytvorenie syntetických dát z už existujúcich ich transformáciou. V prípade transformácie časti existujúcich obrázkov v trénovacej množine do jednokanálovej podoby (grayscale) by modelu farebnosť obrázkov chýbala (absencia hrán by konvolučnej neurónovej sieti neumožnila správne zovšeobecniť informácie o tele vtákov).

¹ nevyváženosť tried sa pokúsime vyriešiť tiež manipulovaním hyper-parametra *stride*. Pre vysvetlenie pozri [6.3. Padding and Stride — Dive into Deep Learning 0.17.5 documentation \(d2l.ai\)](https://d2l.ai/chapter_convolutional_neural networks/6.3.padding-and-stride.html)

Ďalšou (aj keď teoretickou) implikáciou z našich úvah je že klasifikujúca neurónová sieť môže byť menej úspešnejšia pri klasifikácii tých obrázkov z validačnej množiny ktoré sa nenachádzajú v tréningovej množine². V tréningovej množine očakávame čo najrovnomernejšie zastúpenie obrázkov všetkých možných druhov vtákov ktoré môžu byť pri reálnom použití modelu na klasifikáciu vložené. Je teda jasné že ak by sme analyzovali rozloženie hodnôt konkrétnych pixelov v obrázkoch zo všetkých 3 množín nemali by rovnaké rozloženie. Najčastejšou situáciou ktorá nastáva je že rozloženie tréningovej množiny je iné ako rozloženie množiny určenej na validáciu. Takáto situácia nám zároveň logicky dovoľuje porovnávať výkon modelov hlbokého učenia na validačnej množine. Jednou z prvých podmienok ktorú na náš očakávaný klasifikačný model kladieme je že naučený klasifikačný model musí dosahovať nízku hodnotu chyby klasifikácie podľa stanovenej metriky pri jeho tréningu. Výkon viacerých modelov porovnáme podsunutím validačnej množiny. Modely ktoré dostanú na klasifikáciu obrázky vtákov z validačnej množiny zároveň budeme optimalizovať menením hodnôt hyper-parametrov, teda tých parametrov umelej neurónovej siete ktoré sa sieť pri tréningu sama neučí.

Problém klasifikácie vtáčích druhov

Konvolučné neurónové siete predstavujú len jednu z mnohých architektúr ktoré existujú³. Logicky však nastáva otázka prečo ich použiť práve na klasifikáciu obrázkov vtáčích druhov. Historicky základy konvolučných neurónových sietí položil Frank Rosenblatt vo svojej práci opisujúcej biológiu inšpirovanú implementáciu viac-vrstvového plne-prepojeného perceptrónu "*A probabilistic model for information storage and organization in the brain*". V nej opisoval spôsob akým možno takýto perceptrón naučiť identifikovať obrázok z tréningovej množiny. Použitie logického perceptrónu navrhnutého Frankom Rosenblattom by bolo však na náš problém klasifikácie nevhodné, pretože:

- logický perceptrón by ignoroval globálnu informáciu o tvaroch vtákov ktorú potrebujeme získať a abstrahovať.
- takýto logický perceptrón by len obtiažne dokázal identifikovať vtákov na obrázkoch na ktorých by sa nachádzalo mnoho našej úlohe nesúvisiacich objektov.
- logický perceptrón by tiež nedokázal identifikovať vtákov nachádzajúcich sa v rôznych častiach obrázku
- perceptrón by od nás predom vyžadoval určenie vhodných príznakov ktoré odlišujú vtákov. My však chceme určenie týchto príznakov naučiť konvolučnú sieť

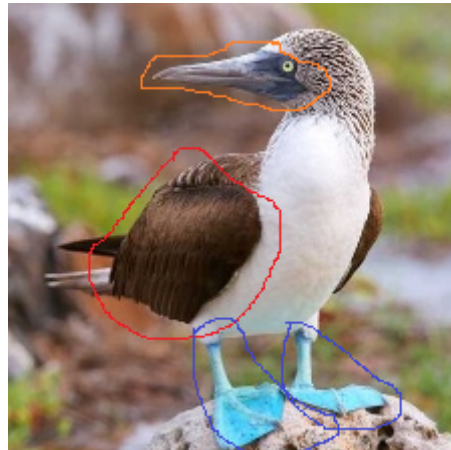
Na vyriešenie problému získania globálnej informácie o tvare vtákov z obrázkov nadviažeme na myšlienku Hubel a Wiesel, pričom podmienkou ktoré na funkciu konvolučnej neurónovej siete kladieme je, že:

- vrstvy umelej neurónovej siete ktoré budú krok za krokom vytvárať abstraktnejší pohľad na tvar vtákov (napr. identifikujú hrany krídel) budeme nazývať konvolučné vrstvy. Tieto vrstvy vytvárajú topografickú mapu.

² Takýto druh vtáku neexistuje, ale vo validačnej množine môže byť iný obrázok vtáku z toho istého druhu

³ <https://www.asimovinstitute.org/neural-network-zoo>

- skupiny perceptrónov budú zodpovedné za identifikáciu čiastkovej informácie: úzke nohy, zobák, oči na odvrátených stranách hlavy.
- skupiny perceptrónov ktoré budú identifikovať inú časť rovnakej časti tela vtáka (napr. ľavú a pravú nohu) budú spojené a skupiny perceptrónov prepojené
- nižšie vrstvy umelej neurónovej siete budú získavať informáciu o samostatných častiach tela vtáka. Stredné vrstvy začnú vytvárať informáciu o zakrivení častí jeho tela (nohy vo vertikálnej polohe). Vyššie vrstvy doplnia informáciu o prepojení častí tela vtáka, ktoré identifikovali jej stredné vrstvy (oranžová + červená + modrá časť blízko seba).



Druh vtáku: Abbotts Booby žijúceho na ostrovoch vo východnej časti Indického oceánu

Konvolučná neurónová sieť však bude len extraktorom informácií pre klasifikátor predstavovaný viacvrstvovým perceptrónom. Konvolučná vrstva sa bude učiť extrahovať informácie o vzhľade vtákov učením sa hodnôt konvulčných jadier. Farebnosť obrázkov, resp. ich zobrazenie vo viacerých kanáloch akým je napr. trojkanálový farebný model RGB je dôležitým faktorom pretože pri jednom kanáli ("grayscale" obrázkoch, inverzných transformáciách farebných obrázkov) by sieť nedokázala nachádzať črty tela vtákov tak dobre ako pri farebných.

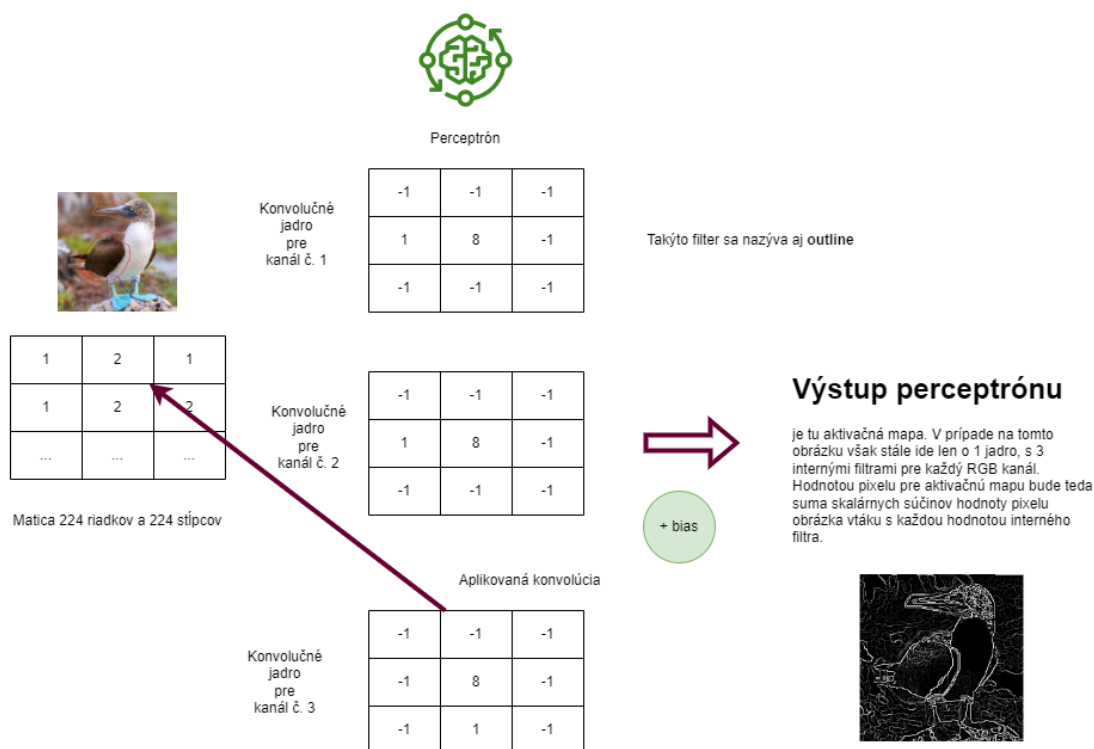
Vstupom pre extraktor informácií o tvare tela vtákov bude pri učení množina 58 388 obrázkov, resp. trojíc matíc o 224 riadkoch a 224 stĺpcoch pre každý kanál farebného modelu RGB. Poskytnutie vstupu extraktoru vo forme matíc je dôležité pre zachovanie priestorovej informácie - hodnôt príznakov pre jednotlivé časti tela.

Odpoveďou na našu predom položenú otázku, prečo je použitie konvulčnej neurónovej siete výhodnejšie oproti použitiu viacvrstvého perceptrónu je teda *"preto, lebo konvulčná neurónová sieť sa nemusí naučiť tvar všetkých druhov vtákov nachádzajúcich sa na všetkých možných miestach na obrázku ale na identifikáciu druhu sa stačí naučiť len abstraktné príznaky správne identifikujúce druh vtáku"*. Konvulčné neurónové siete sú zároveň efektívnejšou implementáciou klasifikujúceho modelu a to aj pre ich konvulčné vrstvy, ktoré vyžadujú menej parametrov⁴ (v porovnaní s plne prepojenými vrstvami) a tiež preto lebo ich výpočty možno ľahko paralelizovať na grafickej karte.

⁴ [6. Convolutional Neural Networks — Dive into Deep Learning 0.17.5 documentation \(d2l.ai\)](#)

Návrh architektúry konvolučnej neurónovej siete

Predtým než začneme učiť našu sieť klasifikovať vtáacie druhy je potrebné ju naučiť vtáacie druhy na obrázkoch identifikovať. Na vykonanie tejto úlohy sa musí naučiť váhy pre filtre - extraktory informácií (inak volané aj konvolučné jadrá). V našom prípade budeme konvolučnú sieť učiť na 2-dimenzionálnych obrázkoch, no konvolučné siete môžu byť použité aj na prácu s 3-dimenzionálnym obrazom. Rozmer konvolučných jadier pre našu sieť určuje počet kanálov vo farebnom modeli. Konvolučné jadrá, resp. filtre budú mať teda rozmer 3x3. Predpokladáme že na to aby bola sieť schopná klasifikovať vtáky musí mať na to výpočtovú silu, čo znamená viac ako 1 perceptrón. Vzhľadom na to že sieť bude pracovať s obrázkami vo farebnom modeli RGB bude každý perceptrón tvorený trojicou konvolučných jadier, resp. filtrov. Každý perceptrón vyprodukuje aktivačnú mapu, ktoréj hodnoty pixelov nazývame aktiváciami. Aktivácie sú výstupy procesu nazývaného konvolúcia (resp. cross-korelácia) ktorý znázorňuje nasledujúci obrázok. Súčasťou experimentov bude tiež výmena inicializátorov váh, započítavanie a nepoužívanie chýb / bias-ov ku každej lineárnej funkcii z perceptrónov:



Konvolúcia na učenie sa tvaru vtákov

sa výrazne od cross-korelácie implementovanej v rámci TensorFlow / Keras nelíši. Stále ide o násobenie hodnôt pixelov z obrázka zodpovedajúcou váhou z konvolučného jadra. Suma zo skalárnych súčinov pre každý pixel je hodnotou výstupu pre daný pixel v aktivačnej mape.

Situácia na obrázku zobrazuje činnosť iba 1 perceptrónu. Týchto perceptrónov na vrstve však býva viac. Počet perceptrónov určuje počet výstupov z vrstvy.

Rozdiel medzi koreláciou a konvolúciou neznamena rozdiel v schopnostiach siete.

Ďalším dôvodom pre ktorý chceme použiť filtre v konvolučnej neurónovej sieti je ich schopnosť naučiť sieť extrahovať príznaky ktoré odlišujú rôzne druhy vtákov od seba. Tieto filtre na prvých aktivačných mapách nie sú ešte schopné zachytiť veľa informácie, no po prechode vstupu viacerými konvolučnými vrstvami sú tieto filtre efektívnejšie práve vďaka “vyčisteniu” aktivačných máp od často nepotrebných informácií.

Experiment č. 1 - Softmax regresia - Klasifikácia druhov vtákov

Zdrojový kód k experimentu	ann/1-experiment.ipynb at main · viktorFIIT/ann (github.com)
Výsledok experimentu	slabo preučенý model

Parametre ktorým riadim proces učenia môžeme rozdeliť do dvoch skupín. Na tie ktoré určujú architektúru siete a váhy filtrov konvolučných vrstiev, a na doplnujúce ktoré určujú napr. počiatočné hodnoty váh⁵ alebo započítanie chýb (bias) do modelu.

Prvá konvolučná vrstva s filtermi o rozmere 3x3⁶ (podľa počtu kanálov v RGB farebnom modeli) obsahuje 32 blokov (perceptrónov) kde každý blok obsahuje trojicu konvolučných filtrov (pre každý kanál). Výber väčších konvolučných filtrov znásobuje potrebné výpočtové zdroje. Zmenšenie rozmeru konvolučných filtrov zase ovplyvňuje množstvo vizuálnej informácie ktorú filter - detektor tela vtáku vo vstupe resp. aktivačnej mape vidí.

```
# lowest convolutional layer for identification of the edges of birds
model.add(Conv2D(32, (3, 3), padding='same', input_shape=SHAPE))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

Obrázok č. 1: Prvá konvolučná vrstva na zachytenie lokálnych príznakov

Pridanie vrstvy pre MaxPooling zvyšuje citlivosť neurónov na skrytých vrstvách na vizuálne príznaky tela vtákov a zároveň znižuje závislosť siete od priestorovej informácie tak že sa bude vedieť rozhodnúť pre daný druh vtáku bez ohľadu na umiestnenie vtáka na obrázku. MaxPooling znižuje závislosť konvolučných vrstiev od priestorovej informácie, tak že prenecháva posledné rozhodnutie na viacvrstvový perceptrón.

⁵ Vyskúšané inicializátory váh sú:

⁶ [6.4. Multiple Input and Multiple Output Channels — Dive into Deep Learning 0.17.5 documentation \(d2l.ai\)](#)

Vzhľadom na to že riešime problém viactriednej klasifikácie sa nám ako najčastejšia možnosť ponúka použiť ako aktivačnú funkciu *Sigmoid*, *Softmax*, *ReLU* alebo *Leaky ReLU*:

- pri výbere aktivačnej funkcie *sigmoid* pre všetky vrstvy treba počítať s jej nevýhodou že saturuje gradient. Pri veľkých hodnotách gradientu sa gradient výrazne nemení a učenie stagnuje.
- pri výbere aktivácii *ReLU* treba počítať s tým, že niektoré neuróny môžu byť inhibované. Túto nevýhodu rieši jej náhrada *Leaky ReLU*.
- *softmax* aktivácia sa používa na viac triednu klasifikáciu na výstupnej vrstve pričom produkuje vierohodnosť / pravdepodobnosť príslušnosti klasifikovaného objektu do jednej z tried.

Výstupom z prvej konvolučnej vrstvy bude aktivačná mapa s 32 kanálmi. Filtre nie sú zoradené paralelne ale sekvenčne. Paralelne zoradené filtre a teda preferovanie širších než hlbších sietí by sa aj podľa autora⁷ vyplatilo skôr pri klasifikácii obrázkov na ktorých by bol vták umiestnený len v ich časti, pričom by sme museli brať do úvahy aj informáciu o ďalších elementoch na obrázku. Toto nám však pri orezaných obrázkoch a vycentrovaných vtákoch na obrázkoch odpadá. Parameter *padding* je pre túto konvolučnú vrstvu a aj všetky ostatné nastavený na hodnotu "same"⁸ čo zároveň pri prihladnutí na veľkosť filtra (3x3 pixely) znamená že konvolučná vrstva nastaví všetky pixely v prvých stĺpcoch zľava aj sprava aktivačnej mapy a zároveň v prvých riadkoch zhora aj dolu aktivačných máp na hodnotu 0. Veľkosť aktivačných máp sa pri tomto nastavení parametra konvolúcií nemení. Vzhľadom na to že pri tomto experimente ešte s veľmi hlbokou neurónovou sieťou nepracujem je dôvodné predpokladať, že aktuálne nastavenie nespôsobí výraznú akumuláciu chyby klasifikácie pri učení (predpokladáme že sieť aj v produkcii dostane vycentrovaný a orezaný obrázok s vtákom). Vzhľadom na to že sa vtáci v tréningovej, validačnej a testovacej množine nachádzajú vždy takmer v strede obrázka predpokladáme že dané nastavenie siete nespôsobí výraznú nepresnosť klasifikácie vtákov (hoci bude sieť nepresná na okrajoch obrázka).

```
model = tf.keras.models.Sequential()

# lowest convolutional layer for identification of the edges of birds
model.add(Conv2D(32, (3, 3), padding='same', input_shape=SHAPE))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

# convolutional layer to learn and store mid-level features of the bird species
model.add(Conv2D(32, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

# highest convolutional layer to store complex information about the look of birds
model.add(Conv2D(32, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

# and finally mlp
model.add(Flatten())
model.add(Dense(64, activation="relu"))
model.add(Dense(Train_Category_count, activation="relu"))
```

Obrázok č. 2: Prvotný návrh klasifikujúcej Konvolučnej neurónovej siete

⁷ [A Simple Guide to the Versions of the Inception Network | by Bharath Raj | Towards Data Science](#)

⁸ Hodnota "same" sa uvádza pri práci s rámcom TensorFlow

Prechodom konvolučnými vrstvami sa zároveň zvyšuje počet kanálov v aktivačných mapách čo je však bežnou praktikou [6.4. Multiple Input and Multiple Output Channels — Dive into Deep Learning 0.17.5 documentation \(d2l.ai\)](#). Fakt že sa nám počet kanálov prechodom cez konvolučné vrstvy zvyšuje nám zaručuje zachytávanie informácie o hranách - lokálne informácie z viacerých kanálov sa spájajú na vytvorenie globálnej informácie o hrane. Tréningový dataset má nerovnomerné zastúpenie samičiek a samcov vtákov (samci ktorých je 85 % sú farebnejší), no cieľom klasifikácie je klasifikovať druhy. Vzhľadom na tento fakt sa však oplatí nastaviť ďalší parameter *stride* a to pre dva dôvody: a) na umelé zníženie vplyvu počtu samcov k samičkám v datasete⁹ a b) na zefektívnenie náročného výpočtu. S aktuálnym nastavením siete tak ako je znázornený na obrázku č. 2 však nedosahujeme želaný výsledok. Po takmer 83 minútach tréningovania modelu sa chyba na validačnej množine (variance) výrazne nemení, no čo je možno ešte horšie podiel úspešných klasifikácií voči všetkým stagnuje na nízkej hodnote.

```
model.compile(optimizer = "adam",
              loss = 'categorical_crossentropy',
              metrics = ['accuracy'])

EPOCHS = 10
BATCH_SIZE = 50 # set up such it enables us to apply batch normalization after activation

history = model.fit(train_data, epochs=EPOCHS, batch_size=BATCH_SIZE, validation_data = validation_data, steps_per_epoch=len(train_data), validation_steps=len(validation_data))
```

Epoch 1/10
1825/1825 [=====] - 1006s 551ms/step - loss: 11.1798 - accuracy: 0.0031 - val_loss: 11.1400 - val_accuracy: 0.0025
Epoch 2/10
1825/1825 [=====] - 1009s 553ms/step - loss: 11.0060 - accuracy: 0.0034 - val_loss: 11.1388 - val_accuracy: 0.0025
Epoch 3/10
1825/1825 [=====] - 1015s 556ms/step - loss: 11.0043 - accuracy: 0.0038 - val_loss: 11.1368 - val_accuracy: 0.0025
Epoch 4/10
1825/1825 [=====] - 969s 531ms/step - loss: 10.8939 - accuracy: 0.0039 - val_loss: 10.8145 - val_accuracy: 0.0025
Epoch 5/10
1815/1825 [=====] - ETA: 5s - loss: 10.6187 - accuracy: 0.0039

Obrázok č. 3: Výstupy učenia CNN v prvom experimente viedli k preučeniu modelu

Na vyriešenie aj tohto problému stopneme tréningovanie modelu ak sa po 5 epochách nedostaneme aspoň na úroveň úspešnosti klasifikácie 0.5, pričom za pokrok pri klasifikácii budeme považovať zvýšenie hodnoty tejto metriky o jednotku (na implementáciu použijeme komponent [EarlyStopping \(keras.io\)](#)):

```
EPOCHS = 10
BATCH_SIZE = 50 # set up such it enables us to apply batch normalization after activation

# stop if there is no big training progress towards good accuracy after 5 epochs
# stop training if the model cannot pass at least 0.5 accuracy after 5 epochs
earlyStopCallback = EarlyStopping(monitor='accuracy', mode='max', verbose=1, patience=5, min_delta = 1, baseline=0.5)

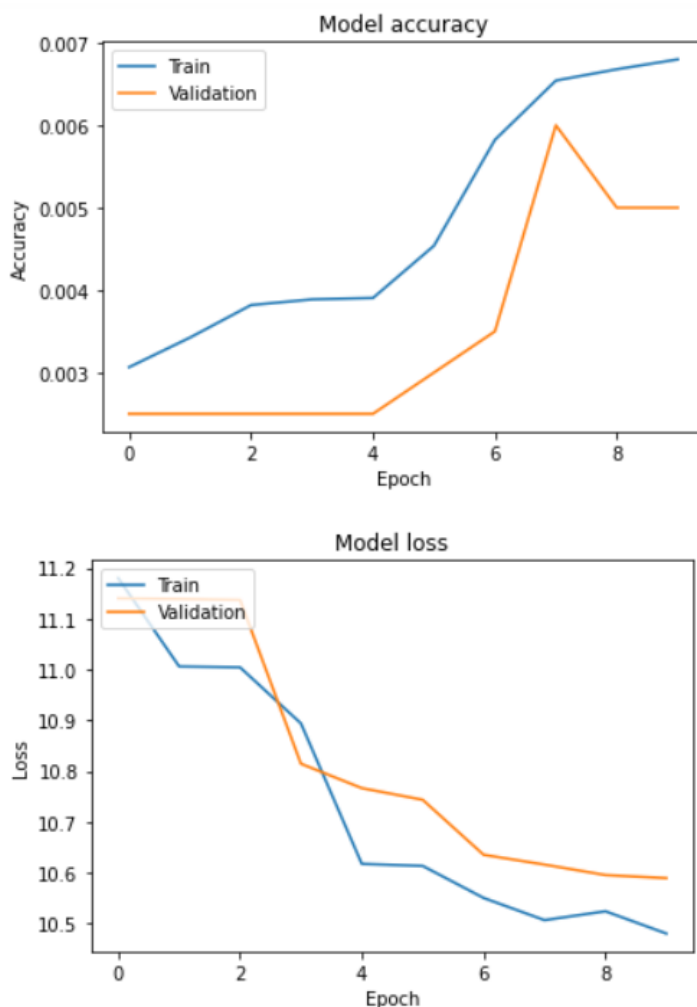
history = model.fit(train_data, epochs=EPOCHS, batch_size=BATCH_SIZE, validation_data = validation_data, steps_per_epoch=len(train_data), validation_steps = len(validation_data), callbacks=[earlyStopCallback])
```

Epoch 1/10
1825/1825 [=====] - 1006s 551ms/step - loss: 11.1798 - accuracy: 0.0031 - val_loss: 11.1400 - val_accuracy: 0.0025
Epoch 2/10
1825/1825 [=====] - 1009s 553ms/step - loss: 11.0060 - accuracy: 0.0034 - val_loss: 11.1388 - val_accuracy: 0.0025
Epoch 3/10
1825/1825 [=====] - 1015s 556ms/step - loss: 11.0043 - accuracy: 0.0038 - val_loss: 11.1368 - val_accuracy: 0.0025
Epoch 4/10
1825/1825 [=====] - 969s 531ms/step - loss: 10.8939 - accuracy: 0.0039 - val_loss: 10.8145 - val_accuracy: 0.0025
Epoch 5/10
1825/1825 [=====] - 961s 526ms/step - loss: 10.6174 - accuracy: 0.0039 - val_loss: 10.7665 - val_accuracy: 0.0025
Epoch 6/10
941/1825 [=====] - ETA: 7:59 - loss: 10.6049 - accuracy: 0.0044

⁹ [6.3. Padding and Stride — Dive into Deep Learning 0.17.5 documentation \(d2l.ai\)](#)

Obrázok č. 4: Nedostatočné učenie CNN je možné obmedziť jeho skorým prerušením

Schopnosť klasifikácie obrázkov vtákov model v súčasnom nastavení zvláda lepšie na (plnej, nezmenšenej) trénovacej množine ako na validačnej, čo sa dá pokladať za očakávaný stav. Model je súčasne slabo preučený, pretože je chyba jeho klasifikácií menšia na trénovacej množine ako na validačných dátach a to takmer počas všetkých epóch - prechodov nad danou množinou.



Obrázok č. 5: Výstupy učenia klasifikácie modelu z 1. experimentu hovoria o jeho preučení

Experiment č. 2 - Softmax regresia - Klasifikácia druhov vtákov

Zdrojový kód k experimentu	ann/1-experiment.ipynb at main · viktorFIIT/ann (github.com)
Cieľ experimentu	oslabiť model z experimentu č. 1

Cieľom tohto experimentu bolo mierne oslabiť a zovšeobecniť informáciu naučenú modelom z 1. experimentu. Na tento účel boli vybrané nasledovné regularizačné techniky:

- normalizácia výstupov z konvolučných vrstiev - vždy po aplikácii max pooling na predprípravu vstupu pre ďalšiu vrstvu vo vyššej kvalite, čo robí úpravou distribúcie dát, na tie s nulovou strednou hodnotou a jednotkovým rozptylom.
- dočasné vynulovanie niektorých neurónov cez techniku dropout.
- zmenšenie dávky zo 70 obrázkov na 50, ako ďalšia regularizačná technika navrhnutá Goodfellow-om.

Poznámky k implementácii

Konvolučná neurónová sieť opísaná v tomto dokumente bola spúšťaná na zariadení s týmito charakteristikami:

Zariadenie	HP Pavilion Gaming Laptop 15
CPU	AMD Ryzen 5 5600H so 6 jadrami
Rámec pre CNN	TensorFlow 2.8.0, Keras 2.8.0
GPU	NVIDIA GeForce RTX 3050 Ti Laptop GPU, architektúra 8.6
Inštalovaný NVIDIA ovládač GPU	GeForce Game Ready Driver pre NVIDIA GeForce RTX 3050 Ti a MS Windows 11, verzia 512.15
Podpora pre TensorFlow GPU	nie, TensorFlow podporuje CUDA 11.2, ktorá nepodporuje Windows 11 https://www.tensorflow.org/install/gpu#software_requirements
CPU Optimalizácia	cez oneAPI Deep Neural Network Library a inštrukčný set AVX AVX2