

## Adat augmentáció PreTrained modell nélkül

### Feladat:

Itt is megmaradtam a kutya cica osztályzásnál, mivel úgyis az adatdúsítást szeretném kísérletezni, de most nem az InceptionV3 pretrained modellel dolgozom.

### A modell felépítése:

1. Képek letöltése, kicsomagolása
2. Alapmodell felépítése amivel még tanítani fogjuk
3. Adat augmentáció
4. Training

Print(model.summary()) paranccsal megnézhető a háló összetétele:

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 295, 295, 32)	2432
batch_normalization (Batch Normalization)	(None, 295, 295, 32)	128
conv2d_1 (Conv2D)	(None, 291, 291, 64)	51264
global_average_pooling2d (Global Average Pooling)	(None, 64)	0
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 128)	8320
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

Itt kitérnék a dropout rétegre aminek a szerepe hogy bizonyos nodeoknál az adatok 50% nem megy tovább a tanításba, ezáltal nem lesznek nagyon elnyomó nodeok. Kicsit másképp megfogalmazva nem egyetlen nagy megfigyelést tesz a rendszer, pl.: a képeken az összes kutya fekete volt ezért a fekete színnel úgymond azonosítja a kutyákat. A modell így sokfajta architektúrát próbál ki és mondhatni jobban általánosít majd. A megjegyzés rovatban a

dokumentáció alján látszik is a végső következtetés. Használva let az úgy nevezett EarlyStopping funkció ahol meg lehet adni egy patience timeot amit a modell epochokban mér és amennyiben a paciencnek megfelelő epochon a modell nem tud javulni leállítja magát, ezzel a túltanulás ellen nem lehet védekezni sajnos. Esetemben ez azért volt szükséges, mert nem let túl jó a modell és nagyon magas epochszámon futtattam, van ahol rengeteg ideig futott de nem javult és ekkor adtam hozzá egy EarlyStoppingot. Továbbá a modell végén visszatöltöm a legjobb modellt, ez mindenképp érdemes megtenni. Amúgy arra nem igazán sikerült rájönnöm a modell miért ilyen gyenge, sokat próbálkoztam az optimalizációval és egy nagyon hasonló hálót használva egy másik modellben egész jó eredményeket értem el itt viszont valami félrement.

A felépített modellben változtattam az aktivációs rétegeket továbbá valahol használtam augmentáció van ahol nem, maga a modell nem let a legjobb de így egész jól látszódott az augmentáció sikeressége. A továbbiakban sok különböző mód futásainak az adatai lesznek összehasonlítva, hogyan tanult a rendszer mennyi a val\_loss és a val\_accuracy. Ez tényleg csak pár adat, de a modellt futtattam vagy 20-30 különböző esetre, ebből csak 4 látszik itt meg az alján egy ami a dropout rétegek nélkül készült el.

## Augmentáció nélkül:

```
100/100 [=====] - 20s 196ms/step - loss: 0.7046 - accuracy: 0.5375 -  
val_loss: 0.6901 - val_accuracy: 0.5000  
Epoch 2/15  
100/100 [=====] - 20s 196ms/step - loss: 0.6896 - accuracy: 0.5495 -  
val_loss: 0.6855 - val_accuracy: 0.5400  
Epoch 3/15  
100/100 [=====] - 20s 197ms/step - loss: 0.6789 - accuracy: 0.5665 -  
val_loss: 0.6801 - val_accuracy: 0.5670  
Epoch 4/15  
100/100 [=====] - 20s 197ms/step - loss: 0.6751 - accuracy: 0.5545 -  
val_loss: 0.6685 - val_accuracy: 0.5770  
Epoch 5/15  
100/100 [=====] - 20s 198ms/step - loss: 0.6915 - accuracy: 0.5410 -  
val_loss: 0.6714 - val_accuracy: 0.5910  
Epoch 6/15  
100/100 [=====] - 20s 198ms/step - loss: 0.6752 - accuracy: 0.5670 -  
val_loss: 0.7939 - val_accuracy: 0.5120  
Epoch 7/15  
100/100 [=====] - 20s 200ms/step - loss: 0.6731 - accuracy: 0.5680 -  
val_loss: 1.3686 - val_accuracy: 0.5040  
Epoch 8/15  
100/100 [=====] - 20s 201ms/step - loss: 0.6725 - accuracy: 0.5360 -  
val_loss: 0.6848 - val_accuracy: 0.5490  
Epoch 9/15  
100/100 [=====] - 20s 200ms/step - loss: 0.6632 - accuracy: 0.5730 -  
val_loss: 0.6601 - val_accuracy: 0.5950  
Epoch 10/15  
100/100 [=====] - 20s 199ms/step - loss: 0.6676 - accuracy: 0.5775 -  
val_loss: 0.6559 - val_accuracy: 0.6070  
Epoch 11/15  
100/100 [=====] - 20s 198ms/step - loss: 0.6601 - accuracy: 0.5905 -  
val_loss: 0.6686 - val_accuracy: 0.5800  
Epoch 12/15  
100/100 [=====] - 20s 197ms/step - loss: 0.6664 - accuracy: 0.5690 -  
val_loss: 0.6668 - val_accuracy: 0.5830  
Epoch 13/15  
100/100 [=====] - 20s 199ms/step - loss: 0.6689 - accuracy: 0.5780 -  
val_loss: 0.6904 - val_accuracy: 0.5410  
Epoch 14/15
```

100/100 [=====] - 20s 197ms/step - loss: 0.6617 - accuracy: 0.5800 -  
val\_loss: 0.6717 - val\_accuracy: 0.5780  
Epoch 15/15  
100/100 [=====] - 20s 198ms/step - loss: 0.6610 - accuracy: 0.5745 -  
val\_loss: 0.6591 - val\_accuracy: 0.5950

Augmentáció nélkül - model.add(Dense(128, activation='sigmoid')):

100/100 [=====] - 20s 195ms/step - loss: 0.7621 - accuracy: 0.5125 -  
val\_loss: 0.7187 - val\_accuracy: 0.5000  
Epoch 2/15  
100/100 [=====] - 20s 197ms/step - loss: 0.7150 - accuracy: 0.5355 -  
val\_loss: 0.7170 - val\_accuracy: 0.5000  
Epoch 3/15  
100/100 [=====] - 20s 198ms/step - loss: 0.6962 - accuracy: 0.5490 -  
val\_loss: 0.6693 - val\_accuracy: 0.5870  
Epoch 4/15  
100/100 [=====] - 20s 199ms/step - loss: 0.6830 - accuracy: 0.5525 -  
val\_loss: 0.6739 - val\_accuracy: 0.5710  
Epoch 5/15  
100/100 [=====] - 20s 200ms/step - loss: 0.6865 - accuracy: 0.5405 -  
val\_loss: 0.6647 - val\_accuracy: 0.5920  
Epoch 6/15  
100/100 [=====] - 20s 201ms/step - loss: 0.6786 - accuracy: 0.5630 -  
val\_loss: 0.6645 - val\_accuracy: 0.5880  
Epoch 7/15  
100/100 [=====] - 20s 202ms/step - loss: 0.6694 - accuracy: 0.5615 -  
val\_loss: 0.6644 - val\_accuracy: 0.5910  
Epoch 8/15  
100/100 [=====] - 20s 202ms/step - loss: 0.6636 - accuracy: 0.5670 -  
val\_loss: 0.6800 - val\_accuracy: 0.5730  
Epoch 9/15  
100/100 [=====] - 20s 201ms/step - loss: 0.6732 - accuracy: 0.5675 -  
val\_loss: 0.6829 - val\_accuracy: 0.5560  
Epoch 10/15  
100/100 [=====] - 20s 201ms/step - loss: 0.6716 - accuracy: 0.5615 -  
val\_loss: 0.6657 - val\_accuracy: 0.5790  
Epoch 11/15  
100/100 [=====] - 20s 202ms/step - loss: 0.6676 - accuracy: 0.5675 -  
val\_loss: 0.6910 - val\_accuracy: 0.5230  
Epoch 12/15  
100/100 [=====] - 20s 202ms/step - loss: 0.6657 - accuracy: 0.5680 -  
val\_loss: 0.6897 - val\_accuracy: 0.5330  
Epoch 13/15  
100/100 [=====] - 20s 203ms/step - loss: 0.6676 - accuracy: 0.5690 -  
val\_loss: 0.6613 - val\_accuracy: 0.5840  
Epoch 14/15  
100/100 [=====] - 20s 204ms/step - loss: 0.6666 - accuracy: 0.5705 -  
val\_loss: 0.7030 - val\_accuracy: 0.5540  
Epoch 15/15  
100/100 [=====] - 20s 202ms/step - loss: 0.6609 - accuracy: 0.5885 -  
val\_loss: 0.6633 - val\_accuracy: 0.5780

Augmentációval(zca\_whitening=True, zoom\_range=0.99, rotation\_range=30) -  
Optimizer = SGD

Epoch 1/15

100/100 [=====] - 49s 486ms/step - loss: 0.6957 - accuracy: 0.5090 -  
val\_loss: 0.6922 - val\_accuracy: 0.5500

Epoch 2/15

100/100 [=====] - 49s 488ms/step - loss: 0.6902 - accuracy: 0.5260 -  
val\_loss: 0.6919 - val\_accuracy: 0.5010

Epoch 3/15

100/100 [=====] - 49s 488ms/step - loss: 0.6893 - accuracy: 0.5340 -  
val\_loss: 0.6873 - val\_accuracy: 0.5460

Epoch 4/15

100/100 [=====] - 49s 487ms/step - loss: 0.6878 - accuracy: 0.5280 -  
val\_loss: 0.6863 - val\_accuracy: 0.5470

Epoch 5/15

100/100 [=====] - 49s 491ms/step - loss: 0.6863 - accuracy: 0.5320 -  
val\_loss: 0.6827 - val\_accuracy: 0.5810

Epoch 6/15

100/100 [=====] - 49s 493ms/step - loss: 0.6768 - accuracy: 0.5390 -  
val\_loss: 0.6701 - val\_accuracy: 0.5920

Epoch 7/15

100/100 [=====] - 49s 489ms/step - loss: 0.6776 - accuracy: 0.5435 -  
val\_loss: 0.6756 - val\_accuracy: 0.5700

Epoch 8/15

100/100 [=====] - 49s 492ms/step - loss: 0.6733 - accuracy: 0.5620 -  
val\_loss: 0.6794 - val\_accuracy: 0.5360

Epoch 9/15

100/100 [=====] - 48s 484ms/step - loss: 0.6760 - accuracy: 0.5440 -  
val\_loss: 0.6613 - val\_accuracy: 0.5900

Epoch 10/15

100/100 [=====] - 48s 482ms/step - loss: 0.6741 - accuracy: 0.5585 -  
val\_loss: 0.6692 - val\_accuracy: 0.5700

Epoch 11/15

100/100 [=====] - 48s 482ms/step - loss: 0.6757 - accuracy: 0.5620 -  
val\_loss: 0.6813 - val\_accuracy: 0.5790

Epoch 12/15

100/100 [=====] - 48s 481ms/step - loss: 0.6682 - accuracy: 0.5795 -  
val\_loss: 0.6546 - val\_accuracy: 0.6090

Epoch 13/15

100/100 [=====] - 48s 481ms/step - loss: 0.6783 - accuracy: 0.5420 -  
val\_loss: 0.6723 - val\_accuracy: 0.5580

Epoch 14/15

100/100 [=====] - 48s 483ms/step - loss: 0.6681 - accuracy: 0.5655 -  
val\_loss: 0.6570 - val\_accuracy: 0.5780

Epoch 15/15

100/100 [=====] - 48s 484ms/step - loss: 0.6626 - accuracy: 0.5810 -  
val\_loss: 0.6516 - val\_accuracy: 0.5920

Megjegyzés: Nagyon lassú lett, valamiért az SGD nem tett jót a modellnek.

**Augmentációval(shear range=0.2, zoom\_range = 0.2, horizontal\_flip = True):**

```
100/100 [=====] - 48s 479ms/step - loss: 0.7028 - accuracy: 0.5315 -  
val_loss: 0.6916 - val_accuracy: 0.5340  
Epoch 2/15  
100/100 [=====] - 48s 476ms/step - loss: 0.6919 - accuracy: 0.5395 -  
val_loss: 0.6905 - val_accuracy: 0.5180  
Epoch 3/15  
100/100 [=====] - 48s 475ms/step - loss: 0.6905 - accuracy: 0.5305 -  
val_loss: 0.6941 - val_accuracy: 0.5030  
Epoch 4/15  
100/100 [=====] - 47s 473ms/step - loss: 0.6862 - accuracy: 0.5460 -  
val_loss: 0.6922 - val_accuracy: 0.5210  
Epoch 5/15  
100/100 [=====] - 47s 473ms/step - loss: 0.6828 - accuracy: 0.5680 -  
val_loss: 0.7421 - val_accuracy: 0.5020  
Epoch 6/15  
100/100 [=====] - 47s 473ms/step - loss: 0.6838 - accuracy: 0.5585 -  
val_loss: 0.6841 - val_accuracy: 0.5390  
Epoch 7/15  
100/100 [=====] - 48s 477ms/step - loss: 0.6692 - accuracy: 0.5765 -  
val_loss: 0.6876 - val_accuracy: 0.5300  
Epoch 8/15  
100/100 [=====] - 48s 479ms/step - loss: 0.6723 - accuracy: 0.5660 -  
val_loss: 0.6757 - val_accuracy: 0.5520  
Epoch 9/15  
100/100 [=====] - 48s 479ms/step - loss: 0.6648 - accuracy: 0.6005 -  
val_loss: 0.6616 - val_accuracy: 0.5940  
Epoch 10/15  
100/100 [=====] - 48s 479ms/step - loss: 0.6639 - accuracy: 0.5895 -  
val_loss: 0.6610 - val_accuracy: 0.6070  
Epoch 11/15  
100/100 [=====] - 50s 496ms/step - loss: 0.6609 - accuracy: 0.5995 -  
val_loss: 0.6566 - val_accuracy: 0.5730  
Epoch 12/15  
100/100 [=====] - 49s 491ms/step - loss: 0.6581 - accuracy: 0.5995 -  
val_loss: 0.6510 - val_accuracy: 0.6130  
Epoch 13/15  
100/100 [=====] - 49s 491ms/step - loss: 0.6564 - accuracy: 0.5860 -  
val_loss: 0.6664 - val_accuracy: 0.5790  
Epoch 14/15  
100/100 [=====] - 48s 483ms/step - loss: 0.6547 - accuracy: 0.6120 -  
val_loss: 0.6876 - val_accuracy: 0.5580  
Epoch 15/15  
100/100 [=====] - 48s 483ms/step - loss: 0.6632 - accuracy: 0.5930 -  
val_loss: 0.6455 - val_accuracy: 0.6270
```

Megjegyzés: Legjobb modell eddig, 3%-ot ver accuracyben azokra amelyek nem lettek augmentálva még többet is legtöbbször és a val\_loss is itt a legkisebb.

## Tapasztalatok:

Fontos az augmentáció, ahol megtehető ott érdemes alkalmazni. A modell szinten mindig ott adta a legjobb eredményeket ahol a képeket fordítottam, kicsit zoomoltam de nem teljesen. Érdemes EarlyStoppingot alkalmazni, a legjobb modellt visszatölteni. A Batch-sizeot sem érdemes 50-100 fölé tenni, mert lassan fog lefutni és legtöbbször nem job eredménnyel. A Dropout alkalmazása is igen hasznos tud lenni, eredmények alapján:

```
Epoch 1/10
50/50 [=====] - 27s 542ms/step - loss: 0.6979 - accuracy: 0.5430 - val_loss: 0.6924 - val_accuracy: 0.5230
Epoch 2/10
50/50 [=====] - 27s 542ms/step - loss: 0.6714 - accuracy: 0.5650 - val_loss: 0.6905 - val_accuracy: 0.5000
Epoch 3/10
50/50 [=====] - 26s 524ms/step - loss: 0.6929 - accuracy: 0.5410 - val_loss: 0.6917 - val_accuracy: 0.5210
Epoch 4/10
50/50 [=====] - 26s 525ms/step - loss: 0.6768 - accuracy: 0.5660 - val_loss: 0.6867 - val_accuracy: 0.5120
Epoch 5/10
50/50 [=====] - 26s 521ms/step - loss: 0.6827 - accuracy: 0.5380 - val_loss: 0.6916 - val_accuracy: 0.5330
Epoch 6/10
50/50 [=====] - 26s 519ms/step - loss: 0.6687 - accuracy: 0.5600 - val_loss: 0.6713 - val_accuracy: 0.5760
Epoch 7/10
50/50 [=====] - 26s 516ms/step - loss: 0.6749 - accuracy: 0.5610 - val_loss: 0.7506 - val_accuracy: 0.5000
Epoch 8/10
50/50 [=====] - 26s 518ms/step - loss: 0.6630 - accuracy: 0.5470 - val_loss: 0.6713 - val_accuracy: 0.5410
Epoch 9/10
50/50 [=====] - 26s 519ms/step - loss: 0.6624 - accuracy: 0.5760 - val_loss: 0.6689 - val_accuracy: 0.5630
Epoch 10/10
50/50 [=====] - 26s 517ms/step - loss: 0.6678 - accuracy: 0.5630 - val_loss: 0.7414 - val_accuracy: 0.5440
```

Majdnem gyengébb eredményt adott mint eddig bármikor ezért érdemes dropoutot alkalmazni.