

# Poređenje različitih arhitektura neuronskih mrež za klasifikovanje rendgenskih slika pluća

## Arhitektura modela 1 (model0.py)

Prva arhitektura sadrži ukupno 34 sloja (Slika 1). Za grupisanje slojeva u jedan model koristimo *Sequential* klasu.

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 128, 128, 16)	448
max_pooling2d (MaxPooling2D)	(None, 64, 64, 16)	0
spatial_dropout2d (SpatialDropout2D)	(None, 64, 64, 16)	0
conv2d_1 (Conv2D)	(None, 64, 64, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32)	0
batch_normalization (BatchNormalization)	(None, 32, 32, 32)	128
spatial_dropout2d_1 (SpatialDropout2D)	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 32)	9248
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 32)	0
batch_normalization_1 (BatchNormalization)	(None, 16, 16, 32)	128
spatial_dropout2d_2 (SpatialDropout2D)	(None, 16, 16, 32)	0
conv2d_3 (Conv2D)	(None, 16, 16, 32)	9248
max_pooling2d_3 (MaxPooling2D)	(None, 8, 8, 32)	0
batch_normalization_2 (BatchNormalization)	(None, 8, 8, 32)	128
spatial_dropout2d_3 (SpatialDropout2D)	(None, 8, 8, 32)	0
conv2d_4 (Conv2D)	(None, 8, 8, 32)	9248
max_pooling2d_4 (MaxPooling2D)	(None, 4, 4, 32)	0
batch_normalization_3 (BatchNormalization)	(None, 4, 4, 32)	128
spatial_dropout2d_4 (SpatialDropout2D)	(None, 4, 4, 32)	0
conv2d_5 (Conv2D)	(None, 4, 4, 32)	9248
max_pooling2d_5 (MaxPooling2D)	(None, 2, 2, 32)	0
batch_normalization_4 (BatchNormalization)	(None, 2, 2, 32)	128
spatial_dropout2d_5 (SpatialDropout2D)	(None, 2, 2, 32)	0
conv2d_6 (Conv2D)	(None, 2, 2, 64)	18496
max_pooling2d_6 (MaxPooling2D)	(None, 1, 1, 64)	0
batch_normalization_5 (BatchNormalization)	(None, 1, 1, 64)	256
spatial_dropout2d_6 (SpatialDropout2D)	(None, 1, 1, 64)	0
flatten (Flatten)	(None, 64)	0
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 64)	4160
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 64)	4160
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 3)	195
=====		

Slika 1 Arhitektura prvog modela

## Rezultati i zapažanja:

- najbolji rezultat koji je postigao ovaj model je sledeći:  
**loss: 0.3908 - accuracy: 0.8255**  
**val\_loss: 0.3891 - val\_accuracy: 0.8498**
- postignut je sa hiperparametrom *learning rate* 0.001, *decay* kod optimizacije je 0.001/epoch (gde je *epoch* broj iteracija kroz ceo skup podataka, koji je setovan na 101). Slika 2 prikazuje grafik promene *loss* i *accuracy* kroz epohe.
- Testni skup podataka sadrži 624 slike i rezultati evaluacije modela su sledeći:  
**loss: 0.6255 - accuracy: 0.8253**

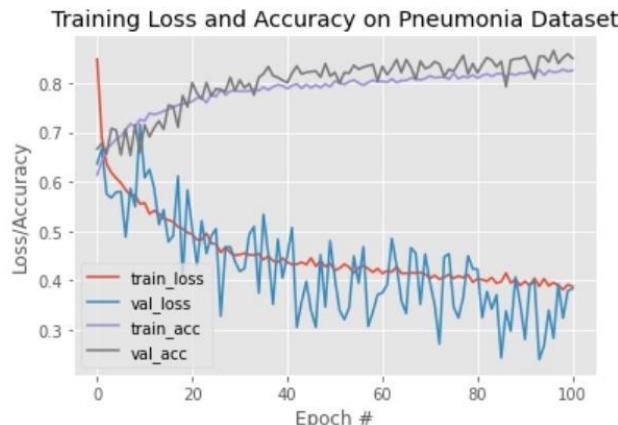
Na osnovu ovih rezultata smatramo da je model dobro obučen.

Neuspeli eksperimenti nad ovim modelom:

- *learning rate* = 0.001, bez *decay*
- *decay* se koristi kako ne bismo morali da stavimo jako male vrednosti *learning rate*-a da *loss* ne bi divergirao
- broj epoha = 101s
- rezultati:

**loss: 0.5192 - accuracy: 0.8011**  
**val\_loss: 0.5487 - val\_accuracy: 0.8162**

Ovaj model nije loš, ali smo pokušavale da smanjimo loss i povećamo accuracy.



Slika 2 Loss/accuracy za prvu arhitekturu modela

## Arhitektura modela 2 (model1.py)

Druga arhitektura (Slika 3) se oslanja na VGG16 mrežu koja je predstavljena 2014. ali se i do danas ima odlične rezultate. Iz nje smo izostavile FC slojeve na vrhu i dodale izmenjenu verziju. Koristi se *average pooling* sloj, *flatten* sloj i dva FC sloja sa regularizacijom između sa 0.5 verovatnoćom. Za optimizaciju koristimo *Adam* jer je trenutno najaktuelniji i najefikasniji, a za funkciju greške koristimo *categorical crossentropy*.

Layer (type)	Output Shape	Param #
input_13 (InputLayer)	[(None, 128, 128, 3)]	0
block1_conv1 (Conv2D)	(None, 128, 128, 64)	1792
block1_conv2 (Conv2D)	(None, 128, 128, 64)	36928
block1_pool (MaxPooling2D)	(None, 64, 64, 64)	0
block2_conv1 (Conv2D)	(None, 64, 64, 128)	73856
block2_conv2 (Conv2D)	(None, 64, 64, 128)	147584
block2_pool (MaxPooling2D)	(None, 32, 32, 128)	0
block3_conv1 (Conv2D)	(None, 32, 32, 256)	295168
block3_conv2 (Conv2D)	(None, 32, 32, 256)	590080
block3_conv3 (Conv2D)	(None, 32, 32, 256)	590080
block3_pool (MaxPooling2D)	(None, 16, 16, 256)	0
block4_conv1 (Conv2D)	(None, 16, 16, 512)	1180160
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block4_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
average_pooling2d_12 (Averag (None, 2, 2, 512)	0	
flatten (Flatten)	(None, 2048)	0
dense_24 (Dense)	(None, 64)	131136
batch_normalization_11 (Bac (None, 64)	0	
dropout_12 (Dropout)	(None, 64)	0
dense_25 (Dense)	(None, 3)	195
<hr/>		
Total params:	14,846,275	
Trainable params:	131,459	
Non-trainable params:	14,714,816	

Slika 3 Arhitektura drugog modela

Rezultati i zapažanja:

Najbolji rezultat koji je ova mreža postigla jeste

loss: 0.2960 - accuracy: 0.8250

val\_loss: 0.2953 - val\_accuracy: 0.8155

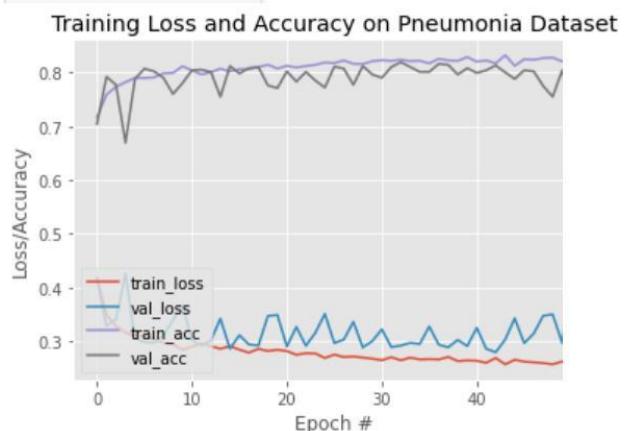
learning rate je u ovom slučaju postavljen na

Slika 4 prikazuje promene loss i accuracy kroz

Testni skup podataka sadrži 624 slika i rezultati

loss: 1.1034- accuracy: 0.7310

Ovaj model se nije pokazao dobro na testnom skupu i zbog toga ne može se smatrati upotrebljivim.



Slika 4 Loss i accuracy drugog modela

Neuspeli eksperimenti:

1. Eksperiment:

- $epoch = 25$ ,  $steps per epoch = 500$
- rezultat: **loss: 0.5171 - accuracy: 0.7521 - val\_loss: 0.4341 - val\_accuracy: 0.8117**

2. Eksperiment:

- $epoch = 100$ ,  $batch size = 64$
- $learning rate = 0.001$
- dodali smo  $decay=learning rate/epoch$
- rezultat: **loss: 0.4251 - accuracy: 0.7945 - val\_loss: 0.5141 - val\_accuracy: 0.7917**

3. Eksperiment:

- $dropout$  promenjen sa 0.5 na 0.25 kako bismo izbegli underfitting
- $learning rate = 0.001$  (bez  $decay$ )
- $epoch = 60$ ,  $batch size = 32$ ,  $steps\_per\_epoch=4000$  //  $batch size$
- rezultat: **loss: 0.1884 - accuracy: 0.8770 - val\_loss: 0.3217 - val\_accuracy: 0.7833**

4. Eksperiment:

- veličina slike promenjena sa 128x128 na 224x224
- $steps per epoch=49$
- rezultat: **loss: 0.5371 - accuracy: 0.7331 - val\_loss: 0.4930 - val\_accuracy: 0.795**

### Arhitektura modela 3 (model2.py)

Treća arhitektura sadrži ukupno 26 slojeva (Slika 5). Za grupisanje slojeva u jedan model koristimo *Sequential* klasu.

Rezultati i zapažanja:

- Nakon 100 epoha ( $batch size$  je 64) rezultat je sledeći (Slika 6):  
**loss: 0.1214 - accuracy: 0.8664 - val\_loss: 0.1983 - val\_accuracy: 0.8279**
- Upotrebom regularizacije u odgovarajućoj meri i povećanjem skupa podataka *data augmentation* tehnikom, i dalje postoji *overfitting* ali je u velikoj meri smanjen u odnosu na prethodne verzije.
- Postoje sličnosti sa prvom arhitekturom, ali u ovom slučaju je korišćen veći *zero padding* i u svakom sledećem konvolucionom sloju se duplira broj filtera (32, 64, 128,...). Takođe, *max pooling* sloj je upotrebljen samo na jednom mestu.
- Testni skup ima 624 slike, slika i rezultat evaluacije modela su sledeći:  
**loss: 0.5817 - accuracy: 0.8077**

Model radi dobro i na testnom skupu, ali nije savršen.

Model: "sequential\_3"

Layer (type)	Output Shape	Param #
=====		
zero_padding2d_9 (ZeroPaddin	(None, 230, 230, 3)	0
conv2d_12 (Conv2D)	(None, 112, 112, 32)	4736
batch_normalization_15 (Batch	(None, 112, 112, 32)	448
activation_12 (Activation)	(None, 112, 112, 32)	0
zero_padding2d_10 (ZeroPaddi	(None, 114, 114, 32)	0
max_pooling2d_3 (MaxPooling2	(None, 56, 56, 32)	0
batch_normalization_16 (Batch	(None, 56, 56, 32)	224
dropout_15 (Dropout)	(None, 56, 56, 32)	0
zero_padding2d_11 (ZeroPaddi	(None, 58, 58, 32)	0
conv2d_13 (Conv2D)	(None, 28, 28, 64)	18496
batch_normalization_17 (Batch	(None, 28, 28, 64)	112
activation_13 (Activation)	(None, 28, 28, 64)	0
dropout_16 (Dropout)	(None, 28, 28, 64)	0
conv2d_14 (Conv2D)	(None, 14, 14, 128)	73856
batch_normalization_18 (Batch	(None, 14, 14, 128)	56
activation_14 (Activation)	(None, 14, 14, 128)	0
dropout_17 (Dropout)	(None, 14, 14, 128)	0
conv2d_15 (Conv2D)	(None, 7, 7, 256)	295168
batch_normalization_19 (Batch	(None, 7, 7, 256)	28
activation_15 (Activation)	(None, 7, 7, 256)	0
dropout_18 (Dropout)	(None, 7, 7, 256)	0
flatten_3 (Flatten)	(None, 12544)	0
dense_6 (Dense)	(None, 512)	6423040
dropout_19 (Dropout)	(None, 512)	0
dense_7 (Dense)	(None, 3)	1539
=====		
Total params:	6,817,703	
Trainable params:	6,817,269	
Non-trainable params:	434	

Slika 5 Arhitektura trećeg modela

## Eksperimenti:

### 1. Eksperiment 1

Pre dodavanja dropout slojeva na više mesta u modelu, rezultati su bili solidni, ali postojao je *overfitting*. Bilo je potrebno povećati regularizaciju. Rezultati su bili sledeći:

loss: 0.1172 - accuracy: 0.9267

val\_loss: 0.4141 - val\_accuracy: 0.8202

### 2. Eksperiment 2

Poslednji dropout sa 0.25 verovatnoćom umesto 0.5, nije postojao nikakav vid povećanja obučavajućeg skupa (*data augmentation*).

loss: 0.0885 - accuracy: 0.9506 - val\_loss:

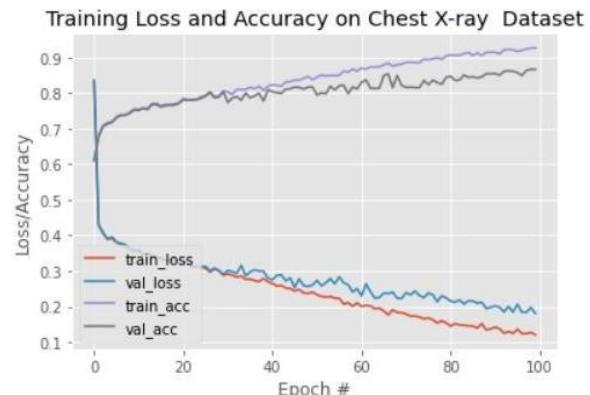
0.4605 - val\_accuracy: 0.8117

### 3. Eksperiment 3

Previše povećani parametri regularizacije, došlo je do loših rezultata obučavajućeg skupa i *underfitting-a*:

loss: 0.3411 - accuracy: 0.7708 - val\_loss:

0.2916 - val\_accuracy: 0.8117



Slika 6 Loss i accuracy trećeg modela