

1 Trenovani

Výsledky kterých bychom chtěli dosáhnout:

- $Top1 = 33.53\%$
- $Top3 = 53.40\%$

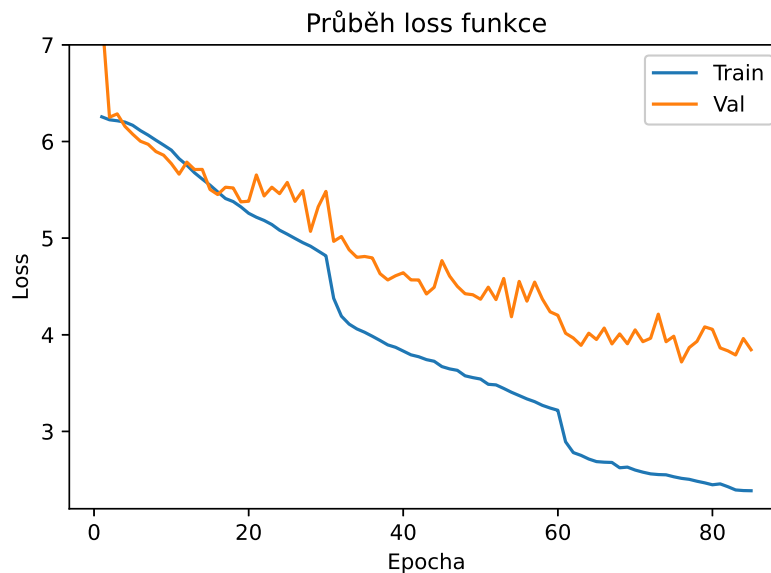
1.1 Z náhodně inicializovaných parametrů neuronové sítě

1.1.1 1. pokus

Trénovací nastavení:

```
groups = 32  
optimizer = 'SGD'  
momentum = 0.9  
lr_scheduler = 'step'  
lr = 0.01  
lr_steps = 30
```

** lr_steps je parametr po kterém se sníží hodnota *learning-rate*(lr) o 10.



Obrázek 1: Průběh změny loss funkce

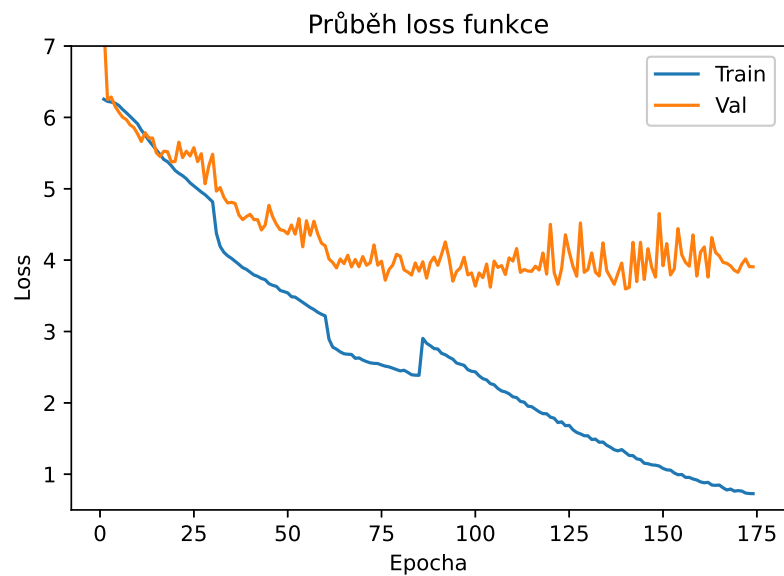
- $Top1 = 23.533\%$
- $Top3 = 41.400\%$
- $Top5 = 49.467\%$

1.1.2 2. pokus

Pokračování trénování z prvního pokusu, tj začíná se trénovat od 85. epochy.

Trénovací nastavení:

```
groups = 32  
start_epoch = 85  
optimizer = 'SGD'  
momentum = 0.9  
lr_scheduler = 'multisteps'  
lr = 0.001  
lr_steps = [95, 130, 160, 200]
```



Obrázek 2: Průběh změny loss funkce

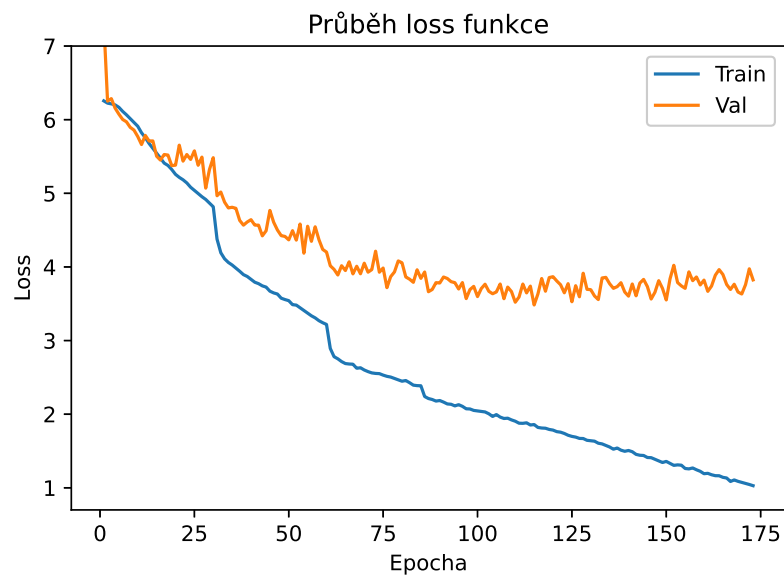
- $Top1 = 31.800\%$
- $Top3 = 50.467\%$
- $Top5 = 59.333\%$

1.1.3 3. pokus

Pokračování trénování z prvního pokusu, tj začíná se trénovat od 85. epochy.

Trénovací nastavení:

```
groups = 32  
start_epoch = 85  
optimizer = 'SGD'  
momentum = 0.9  
lr_scheduler = 'multisteps'  
lr = 0.0001  
lr_steps = [120, 150, 190, 200]
```



Obrázek 3: Průběh změny loss funkce

- $Top1 = 29.000\%$
- $Top3 = 47.733\%$
- $Top5 = 56.133\%$

1.2 Z předtrénovaného modelu *Kinetics400*

1.2.1 1. pokus

Trénovací nastavení:

groups = 32

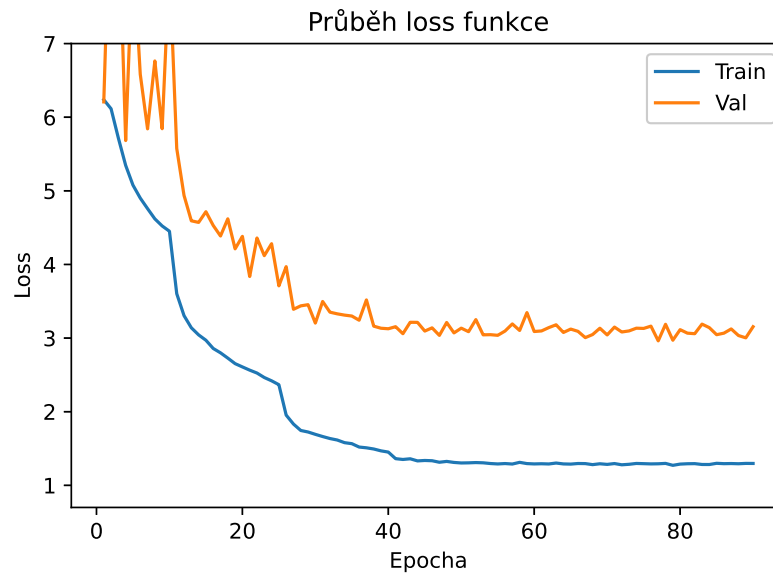
optimizer = 'SGD'

momentum = 0.9

lr_scheduler = 'multisteps'

lr = 0.01

lr_steps = [10, 25, 40, 50, 60, 70, 80, 90, 100]



Obrázek 4: Průběh změny loss funkce

- $Top1 = 33.667\%$
- $Top3 = 54.133\%$
- $Top5 = 62.867\%$

1.2.2 2. pokus

Trénovací nastavení:

groups = 32

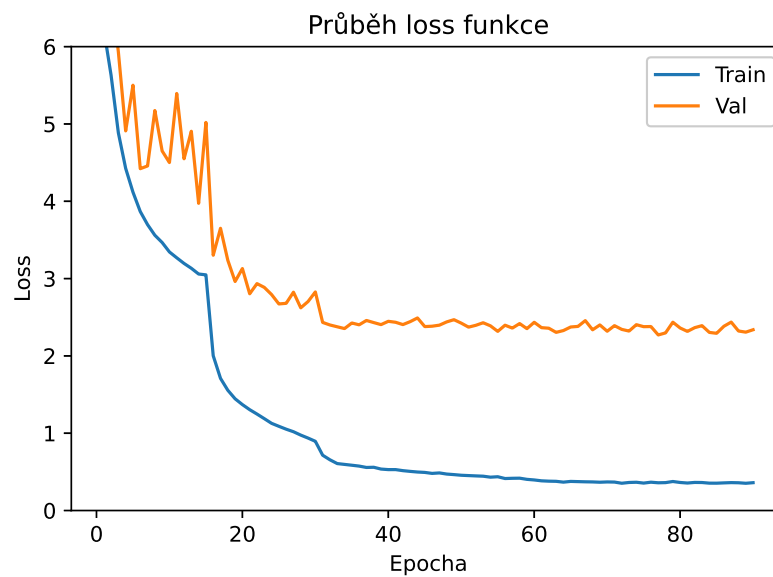
optimizer = 'SGD'

momentum = 0.9

lr_scheduler = 'multisteps'

lr = 0.01

lr_steps = [15, 30, 60, 90, 110]



Obrázek 5: Průběh změny loss funkce

- $Top1 = 50.333\%$
- $Top3 = 72.267\%$
- $Top5 = 77.800\%$

1.2.3 3. pokus

Trénovací nastavení:

groups = 32

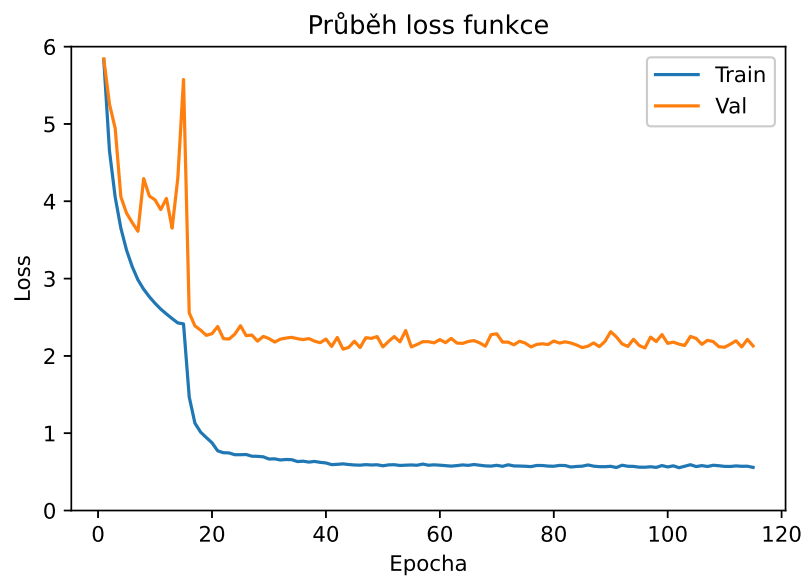
optimizer = 'SGD'

momentum = 0.9

lr_scheduler = 'multisteps'

lr = 0.01

lr_steps = [15, 20, 40, 80, 110]



Obrázek 6: Průběh změny loss funkce

- $Top1 = 54.600\%$
- $Top3 = 76.067\%$
- $Top5 = 81.733\%$

1.2.4 4. pokus

Trénovací nastavení:

groups = 32

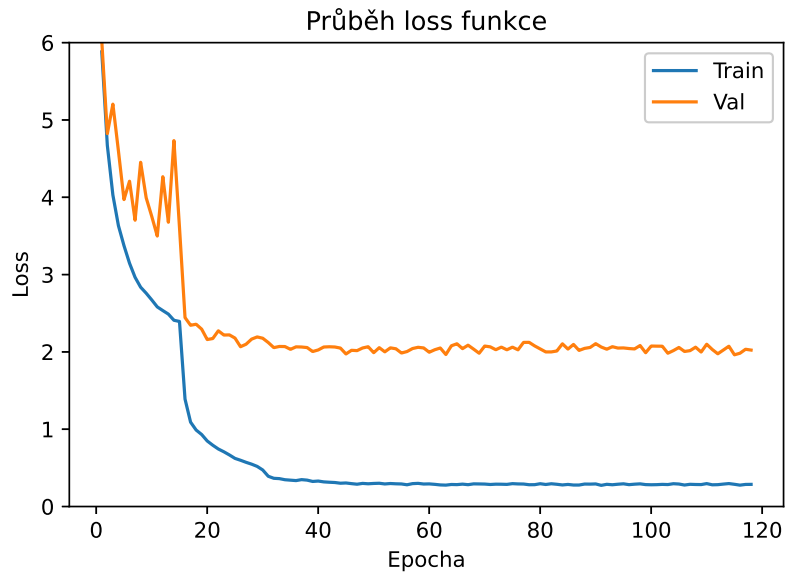
optimizer = 'SGD'

momentum = 0.9

lr_scheduler = 'step'

lr = 0.01

lr_steps = 15



Obrázek 7: Průběh změny loss funkce

- $Top1 = 59.467\%$
- $Top3 = 78.867\%$
- $Top5 = 84.6\%$

** To znamená že mám větší *Accuracy* než výsledky z originálního paperu k datasetu **HAA500** pro rgb I3D model. Dosáhl jsem i vyššího rozpoznávání než je uvedené na paperwithcode, kde jsou uvedené hodnoty *Accuracy* pro I3D three-stream model (RGB, Optical flow, Pose).

2 Zhodnocení? Povídání na konec

Během trénování skript vyhodnocoval nejlepší model a ten uložil, po každé epoše pak porovnával výsledky nového modelu s nejlépe uloženým. Při testování jsem použil právě ten nejlépe vyhodnocený model, ale už jsem zpětně nedokázal dohledat při jaké epoše byl model uložen. Dalo by se to odhadnout zhruba z časů uložení, nevím jestli je to důležité pak v práci zmínit po jaké epoše používáme model...

Zkoušel jsem i jiné optimizery jako Adam, nebo AdamW, ale síť se s nimi vůbec netrénovala, proto používám pouze SGD. Většinu parametrů v args jsem nechal defaultní, mám zkusit případně nějaký z nich pozměnit? Případně mám nějak pozměnit parametry learning rateu, abych dosáhl lepších hodnot?