

# 1 Metodologia chyb

Na vstupe su trajektórie buniek ako 3D tenzory.  $Y$  je požadovaná,  $\hat{Y}$  je predpovedaná sieťou.

**Pre úplnosť poznámka** o tvare a indexovaní tenzorov  $Y$  a  $\hat{Y}$ :

- rozmer tenzora je  $Y[depth][height][width]$ ,
- indexovanie  $Y[k][j][i]$
- index  $k$  určuje bunku,  $k = \langle 0, depth \rangle$ , v experimente  $k = \langle 0, 38 \rangle$
- index  $j$  určuje časový krok,  $j = \langle 0, height \rangle$ , v experimente  $j = \langle 0, 9057 \rangle$
- index  $i$  určuje súradnicu polohy,  $i = \langle 0, width \rangle$ , v experimente  $i = \langle 0, 3 \rangle$

**Najprv sa definuje chyba** ako rozdiel požadovanej a predpovedanej hodnoty

$$E = Y - \hat{Y} \quad (1)$$

**Vypocet relativnej chyby** je nasledovny : v L1 norme pomer chyboveho tenzora  $E$  a prislusnej dlzky tenzora  $Y$ , dostaneme pomocny tenzor  $r_t$ . Tenzor  $\epsilon$  ma nepatrnu kladnu hodnotu a zabranuje deleniu nulou. Nakoniec sa hodnoty spriemeruju a vynasobia 100.

$$r_t = \frac{|E|}{|\hat{Y}| + \epsilon} \quad (2)$$

$$relative\_error = \bar{r}_t 100\% \quad (3)$$

**Pre modelovanie rozloženia pravdepodobnosti** chyby, normalnym rozdelenim je potrebne mat priemer chyby a rozptyl. Z pythonu som pouzil hotove funkcie

- $E.mean()$ , <https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.mean.html>
- $numpy.std(E)$ , <https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.std.html>

$$mean = E.mean() \quad (4)$$

$$sigma = numpy.std(E) \quad (5)$$

Tu neviem ci to vrati  $\sigma$  alebo  $\sigma^2$ .

**Dalsia vhodna metrika** je root mean square error (RMS). Pocitana ako odmocnina z priemeru druhych mocnin chyb. [https://en.wikipedia.org/wiki/Root\\_mean\\_square](https://en.wikipedia.org/wiki/Root_mean_square) Indexovanie  $\alpha$  je len formalne zjednodusenie - aby sa nemuseli pisat 3 sumy cez i,j,k. Hodnota  $N$  je potom  $N = width * height * depth$ .

$$rms = \sqrt{\frac{1}{N} \sum_{\alpha=1}^N E_{\alpha}^2} \quad (6)$$

v pythone ako

```
rms = numpy.sqrt(numpy.mean(numpy.square(error)))
```

**Dalsia vhodna metrika** je absolutna priemerna chyba. Pocitana ako priemer absolutnych hodnot chyby  $E$  v L1 norme.

$$ams = \frac{1}{N} \sum_{\alpha=1}^N |E_{\alpha}| \quad (7)$$

v pythone ako

```
ams = numpy.mean(numpy.absolute(error))
```

## 2 Spočítané výsledky

Tabuľka uvádza chyby pre siete 0 až 7. Chyby majú priamu interpretáciu ako chyba polohy [um]. Zeleným sú v danej metrike znázornené najlepšie a červeným najhoršie výsledky.

ID	error mean [um]	error sigma [um]	rms [um]	ams [um]	relative_error [%]
0	-8.937	37.229	38.287	14.473	14.681
1	-33.725	205.573	208.321	41.814	42.588
2	-3.513	22.089	22.367	9.055	12.907
3	-0.292	15.36	15.363	6.942	11.321
4	-1.151	10.928	10.988	3.824	7.701
5	-0.224	10.765	10.767	4.248	7.95
6	-0.777	11.711	11.736	4.156	8.428
7	-1.526	11.373	11.474	3.735	7.556

## 3 Je treba

Treba overiť či je to správne. Z každého pohľadu : metodológia, vzorce, interpretácia aj programovanie.