

Neurónové siete Z1

Prvé semestrálne zadanie z predmetu Neurónové siete 29.3.2023.

Radovan Cyprich, Adam Melicher +

▼ Boston Housing

Cieľom zadania je predikovať hodnotu cien domov pomocou daných vlastností pomocou lineárnej regresie.

▼ Popis dát

Tento súbor údajov obsahuje 506 záznamov s 13 atribútmi. Nebolo nutné robiť transformácie keďže už boli v stave s ktorým sme mohli priamo pracovať. Nemuseli sme teda robiť veci ako One-hot encoding.

Pomocou heatmapy sme zistili medzi, ktorými atribútmi sú najvyššie korelácie a ich počet sme zvolili ako vstupné parametre do neurónovej siete.

Zvolené atribúty: "LSTAT", "PTRATIO", "INDUS", "TAX", "MEDV", "RM", "RAD"

Trénovacie a testovacia dáta sme rozdelili v pomere 80 ku 20.

▼ Sieť

Náš model obsahoval:

- vstupnú vrstvu s 6 neurónmi (6 je počet vstupných atribútov).
- 3 skryté vrstvy s aktivačnou funkciou RELU
- výstupnú vrstvu s 1 neurónom

Na výpočet loss-u využívame funkciu `MeanSquaredError`. Na vyhodnotenie presnosti modelu používame `RootMeanSquaredError` a `MeanSquaredError` (RMSE a MSE).

▼ Loss funkcie

MAE

Inými slovami, chcete vedieť, ako blízko sú predpovede v priemere k skutočnému modelu. Nízke hodnoty MAE naznačujú, že model predikuje hodnoty správne. Väčšie hodnoty MAE naznačujú, že model nie je vhodný na predikciu.

RMSE

Ak model nadhodnotil alebo podhodnotil niektoré hodnoty v predikcii mali by sme použiť RMSE.

RMSE je populárna hodnotiacia metrika pre regresné problémy, pretože nielenže vypočítava, ako blízko je predpoveď v priemere skutočnej hodnote, ale tiež naznačuje vplyv veľkých chýb. Veľké chyby budú mať vplyv na výsledok RMSE.

▼ Testovanie

▼ Pytorch vs. Tensorflow

Testovali sme porovnanie implementácie toho istého modelu s rovnakými parametrami a architektúrou. Z grafov je vidieť, že pytorch implementácia má nízke hodnoty, no nevedeli sme prísť na to, že prečo. Pravdepodobne sa jedná o chybu v implementácii alebo výpočtoch týchto hodnôt, i keď je veľmi nepravdepodobné, že by bol výpočet zlý na všetkých troch hodnotách.

Taktiež v pytorch implementácii, po približne 15 epochách sa model javí ako natrénovaný, čo je vysoko nepravdepodobné.

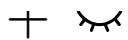




diff only	pytorch	tensorflow
META		
runtime	10s	4m 16s
CONFIG		
BATCH_SIZE	25	25
EPOCHS	250	250
LEARNING_RATE	0.001	0.001
layer sizes	[6,128,32,10]	[6,128,32,10]

Import panel

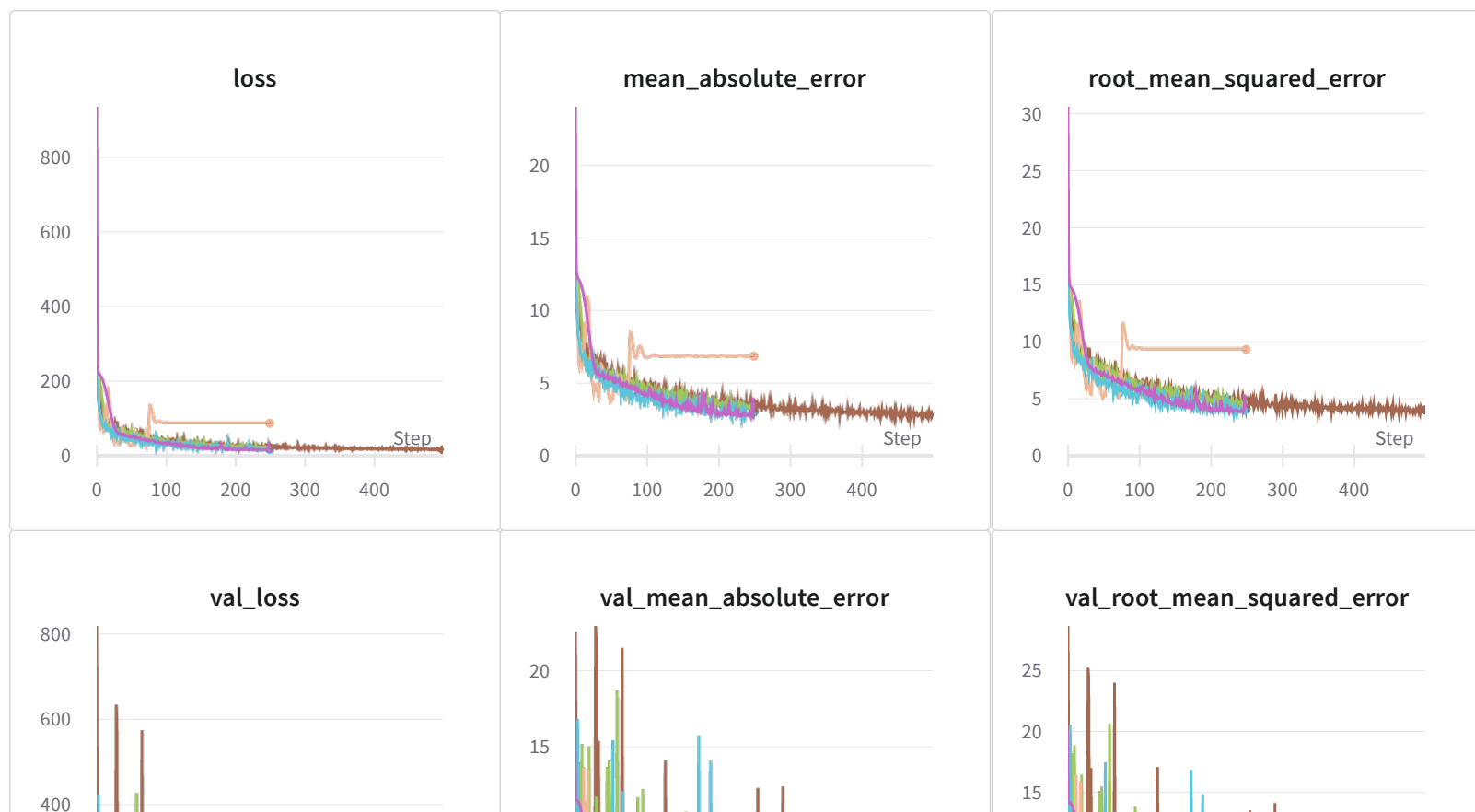
Add panel

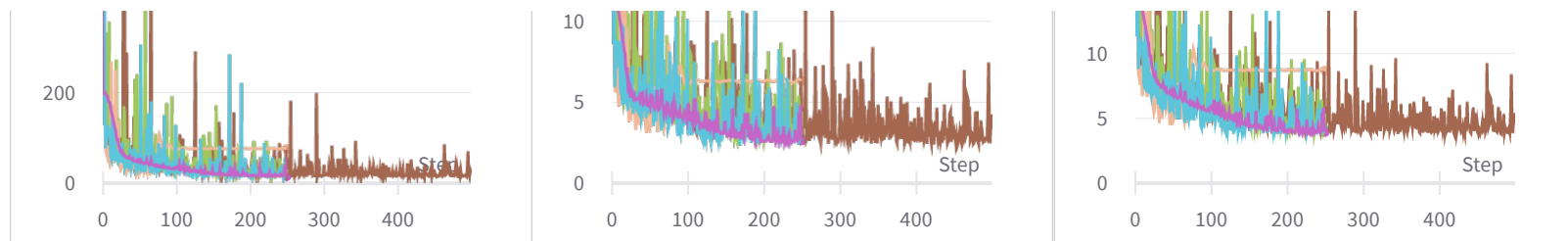


Pre ďalšie testovanie sme teda vybrali implementáciu tensorflow, aj keď beh trvá dlhšie.

▼ Optimizér Adam vs. RMSProp

Pri pokusoch s RMSProp optimizérom sme mali veľmi nepravidelné hodnoty a celkovo boli veľké skoky najmä pri validácii. To je pravdepodobne zapríčinené tým, že tento optimizér robil priveľké skoky na gradiente a nevedel sa dostať plynule nižšie. Toto nebolo odstránené ani po zdvojnásobení počtu epoch. Taktiež pozorujeme silne kladnú koreláciu medzi parametrom momentum a metrikou RMSE (aj ostatnými).





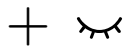
Parameter importance with respect to ||| root_mean_sq... ▾

Parameters 1-2 ▾ of 2 < >

Config parameter	Importance ⓘ ▾	Correlation
momentum		
EPOCHS		

Import panel

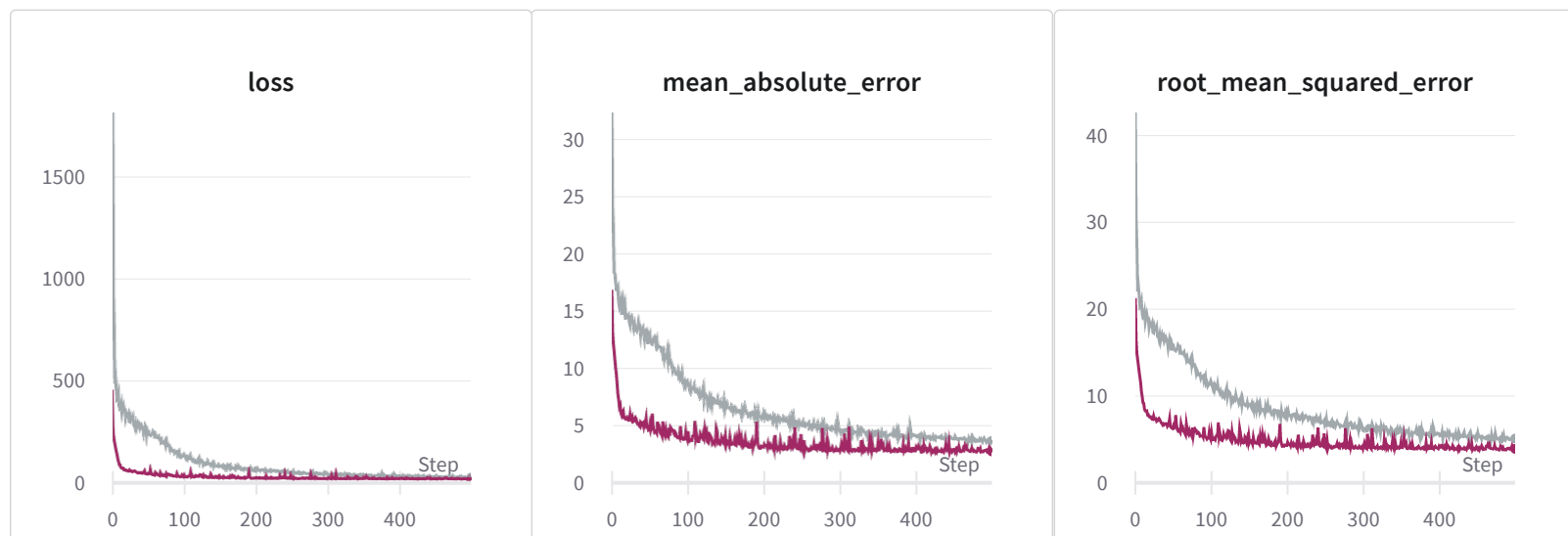
Add panel

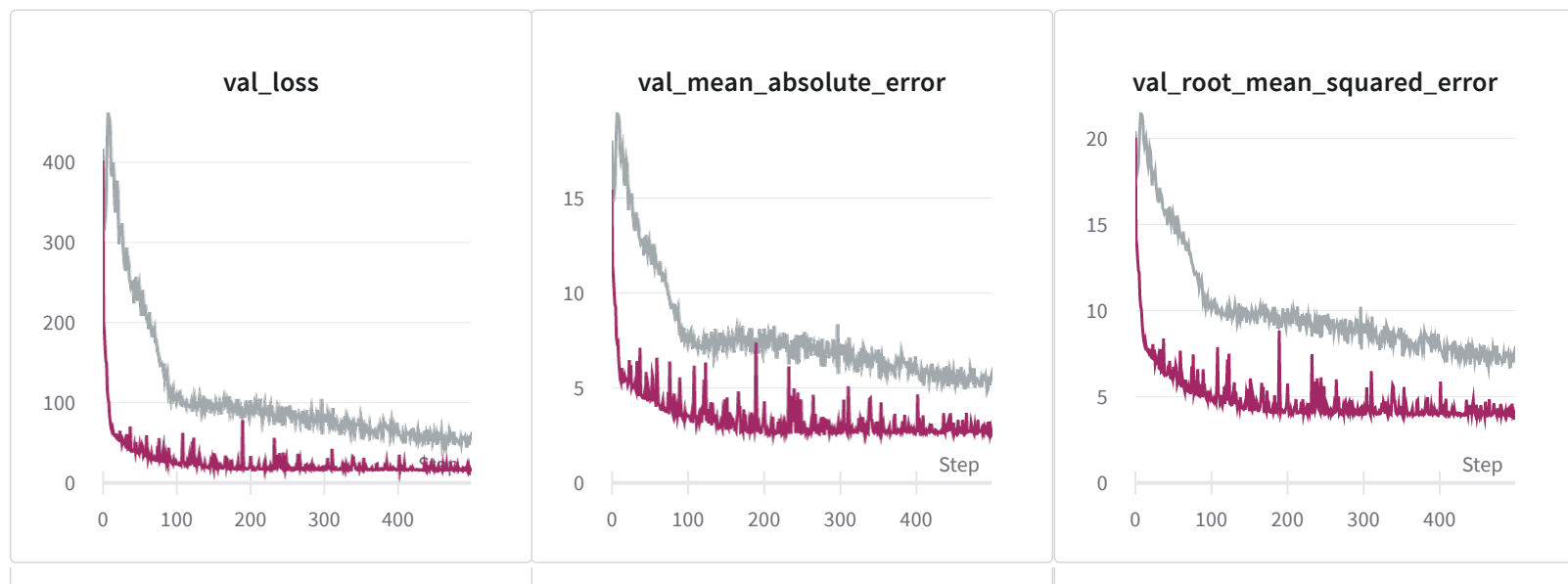


Z grafov jasne vyplýva že lepší optimizér pre náš model je Adam, preto ďalej budeme pokračovať s optimizérom Adam.

▼ Test s dropout vrstvami

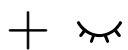
Pridali sme Dropout vrstvu za každú skrytú vrstvu, a dropout rate sme nastavili na 0.5. Na výsledkoch je vidieť, že sme nedosiahli lepšie výsledky, práve naopak horšie, a tréňovanie modelu malo pomalší priebeh.





Import panel

Add panel

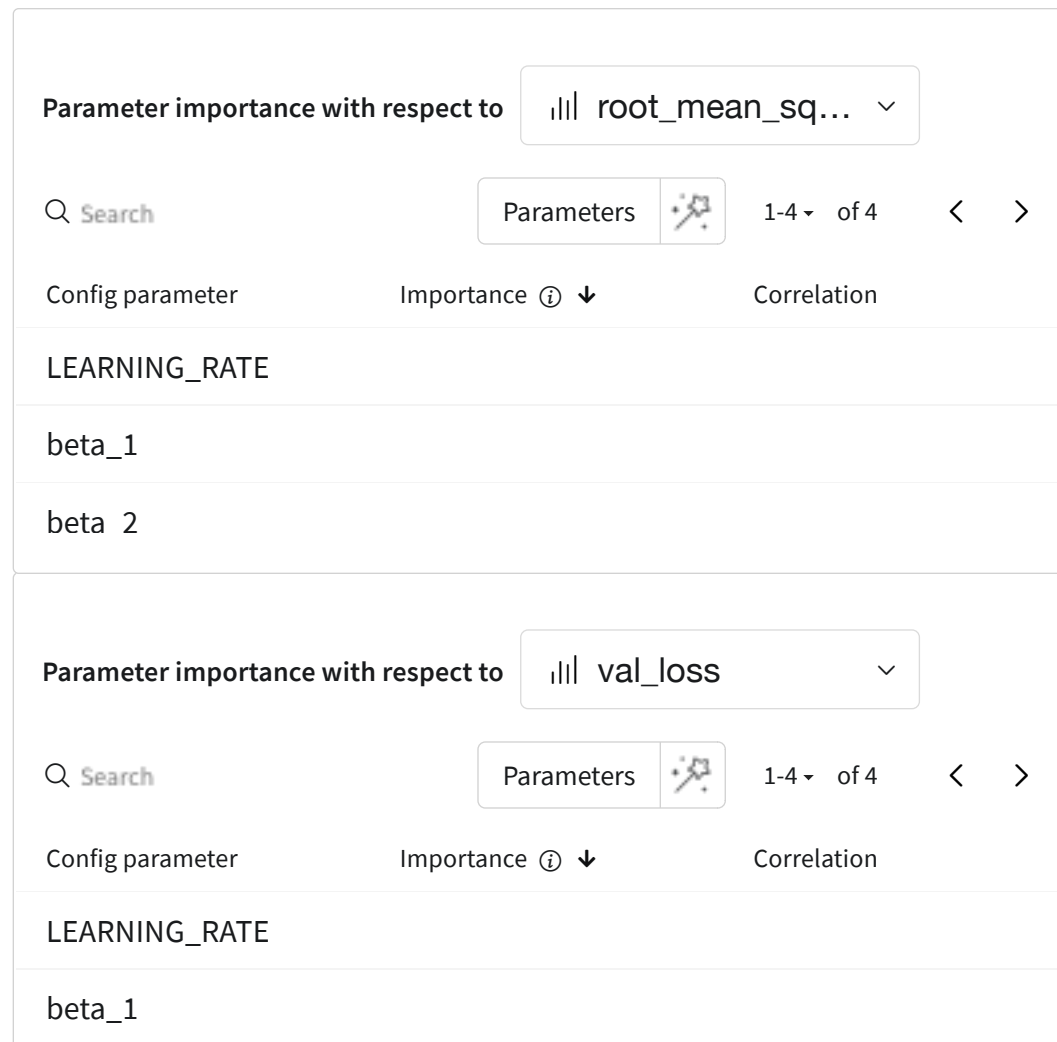


Ďalej teda budeme pokračovať bez Dropout vrstiev

▼ Tuning hyperparametrov

Na nájdenie najlepšej kombinácie parametrov sme použili gridsearch. Vidíme, že na minimalizáciu `valid_loss` a aj RMSE nám najväčšiu úlohu zohrávajú parametre `Learning rate`, a `beta1`. Podľa toho sme vybrali model s konfiguráciou

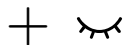
`beta1 = 0.99`, `beta2 = 0.999`, `learning_rate = 0.01`



beta 2

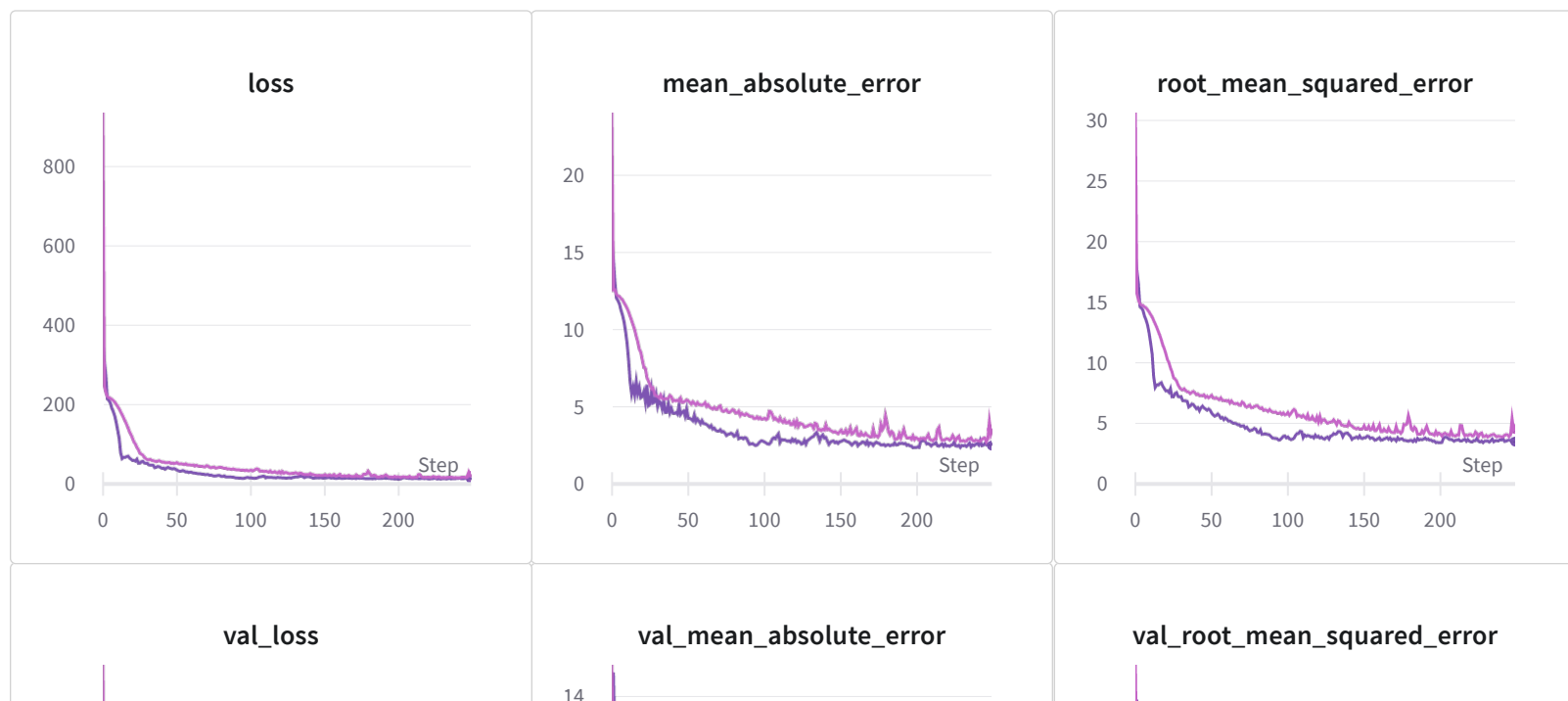
Import panel

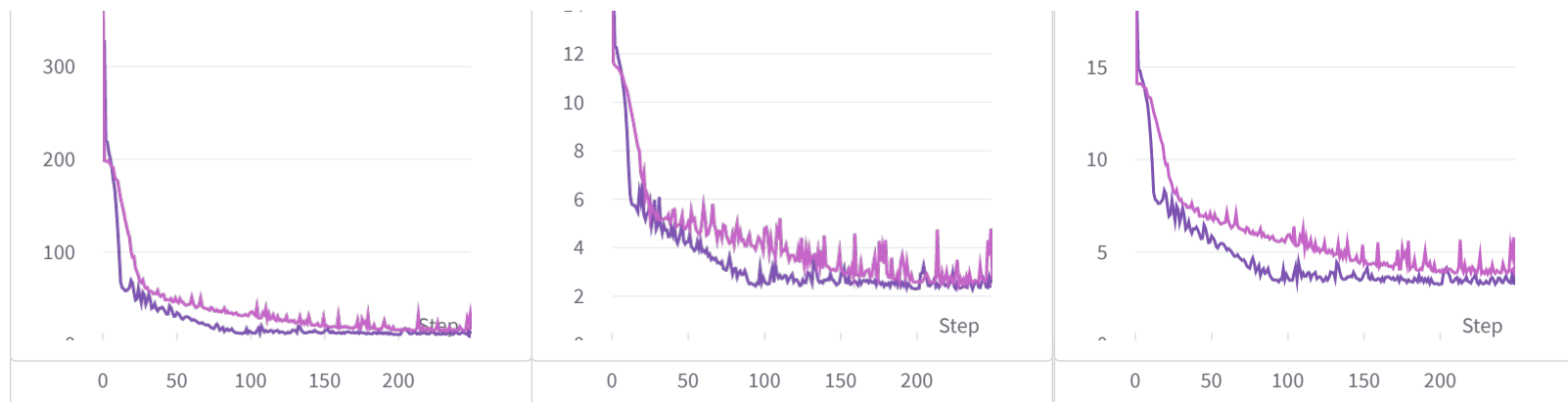
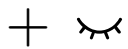
Add panel



▼ Najlepší model

Ako bolo spomenuté vyššie, model s najlepšimi parametrami bol $\text{beta1} = 0.99$, $\text{beta2} = 0.999$, $\text{learning_rate} = 0.01$. Na grafoch vidíme, že sa nám podarilo vylepšiť krivku učenia a aj krivku výsledných metrík. Na každej sme dosiahli zlepšenie.



[Import panel](#)[Add panel](#)

Následne sme model aj testovali na testovacích dátach, ktoré nevidel, a dosiahli sme nasledovné výsledky:

RMSE: 3.9709

MAE: 2.5877

Predikcie: 23.0, 32.4, 15.1, 22.7, 12.6

Dáta: 23.6, 32.4, 13.6, 22.8, 16.1

▼ Záver

V porovnaní jednotlivých frameworkov PyTorch / Tensorflow je jasné, že komplexita práce, prípravy a spracovania dát je v prípade PyTorchu-u omnoho vyššia, nakoľko tento framework je na nižšej úrovni oproti Tensorflow-u. Po testovaní a optimalizovaní hyperparametrov sa nám podarilo dosiahnuť predikcie porovnateľné skutočným hodnotám.