

PyTorch

Anton Rusňák, František Gič

Parkinson

Projekt je zameraný na binárnu klasifikáciu dát ohľadom Parkinsonovej choroby. Binárnou klasifikujeme určujeme jedincov na základe atribútov frekvencie ich hlasu do tried zdravý/chorý.

🔗 ▼ Popis dát

Použitý dataset: <https://archive.ics.uci.edu/ml/datasets/parkinsons>

V dátach sa nachádza 195 záznamov. Dáta nebolo nutné predspracovávať, nakoľko už boli v stave s ktorým sa dalo priamo pracovať. Nebolo teda potrebné robiť One-hot encoding.

Trénovacie a testovacia dáta sme rozdelili v pomere 70 ku 30.

Sieť

Náš model obsahoval:

- vstupnú vrstvu s 22 neurónmi (22 je počet vstupných atribútov).
- 1 skrytú vrstvu s aktivačnou funkciou RELU
- výstupnú vrstvu s 1 neurónom (binárna klasifikácia) a aktivačnou funkciou `Sigmoid`.

Na výpočet loss-u využívame funkciu `BinaryCrossEntropy`. Na vyhodnotenie presnosti modelu používame accuracy.

Využívali sme 2 optimizery - `RMSProp` a `Adam`.

Optimizer Adam

Pre výsledky v reporte sme si vybrali 3 konkrétne behy s týmto optimizerom.

Medzi spoločné črty týchto behov je rovnaká hodnota learning rate-u - 0.001.

Adam 1

Tomuto modelu bol nastavený batch size pomerne nízko - 64, čo spôsobilo, že krivka grafu loss má skokovitú tendenciu - pretože sme lossy jednotlivých batchov priemerovali, čo sa nám zobrazilo aj na grafe, na rozdiel od toho, keď sa použil batch o veľkosti celého datasetu.

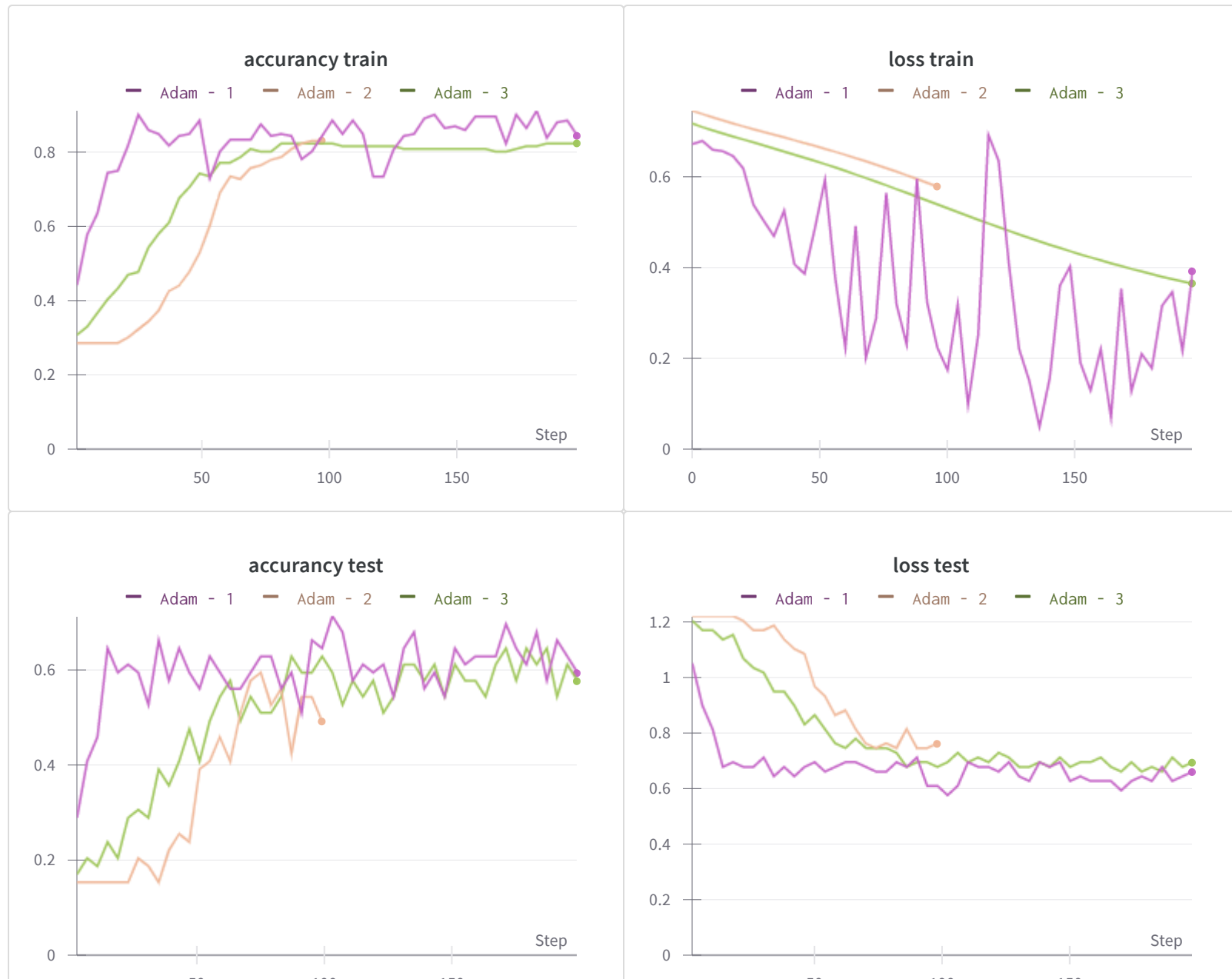
Počet epôch bol 50.

Adam 2 a 3

Po testovaní s malým batch sizom, sme sa rozhodli zvýšiť batch size na veľkosť datasetu - 195. Na krivkách je oproti Adam 1 vidno, že krivka je plynulejšia. Rozdiel medzi týmito dvoma modelmi - Adam 2 a 3 bol len v počtoch epôch. Adam 2 mal nastavený počet epôch 25, a Adam 3 - 50.

Na krivke lossu aj trénovacej, aj testovacej sade môžeme vidieť klesajúcu tendenciu. Oba modely síce začali s nízko inicializovanými váhami, avšak na krivkách presnosti vidíme že sa počas prvých desiatok epôch vyšplhali na vyššie hodnoty.

Vidíme, že Adam 2 - bol vplyvom nízkeho počtu epôch nedotrénovaný - vidíme že loss toho istého modelu Adam 3, len s vyšším počtom epôch, ešte klesal. Avšak hodnotách accuracy boli oba modely približne rovnako.



Optimizer RMSProp

Spoločným znakom jednotlivých modelov je rovnaký learning rate a batch size, nakoľko už pri Adam sme zistili že väčší je lepší (that's what she said).

RMSProp 1

Prvým z modelov je model s nulovou hodnotou moment-(a? -uma? :D). Krivka accuracy dosahovala najnižšie výsledky spomedzi všetkých trénovaných modelov, a loss klesal veľmi pomaly.

RMSProp 2 a 3

Týmto dvom modelom sme nastavili momentum na hodnotu 0.5. Rozdiel medzi týmito dvomi modelmi je v počte epôch - 3 mala skrátené epochy z 50 na 25.

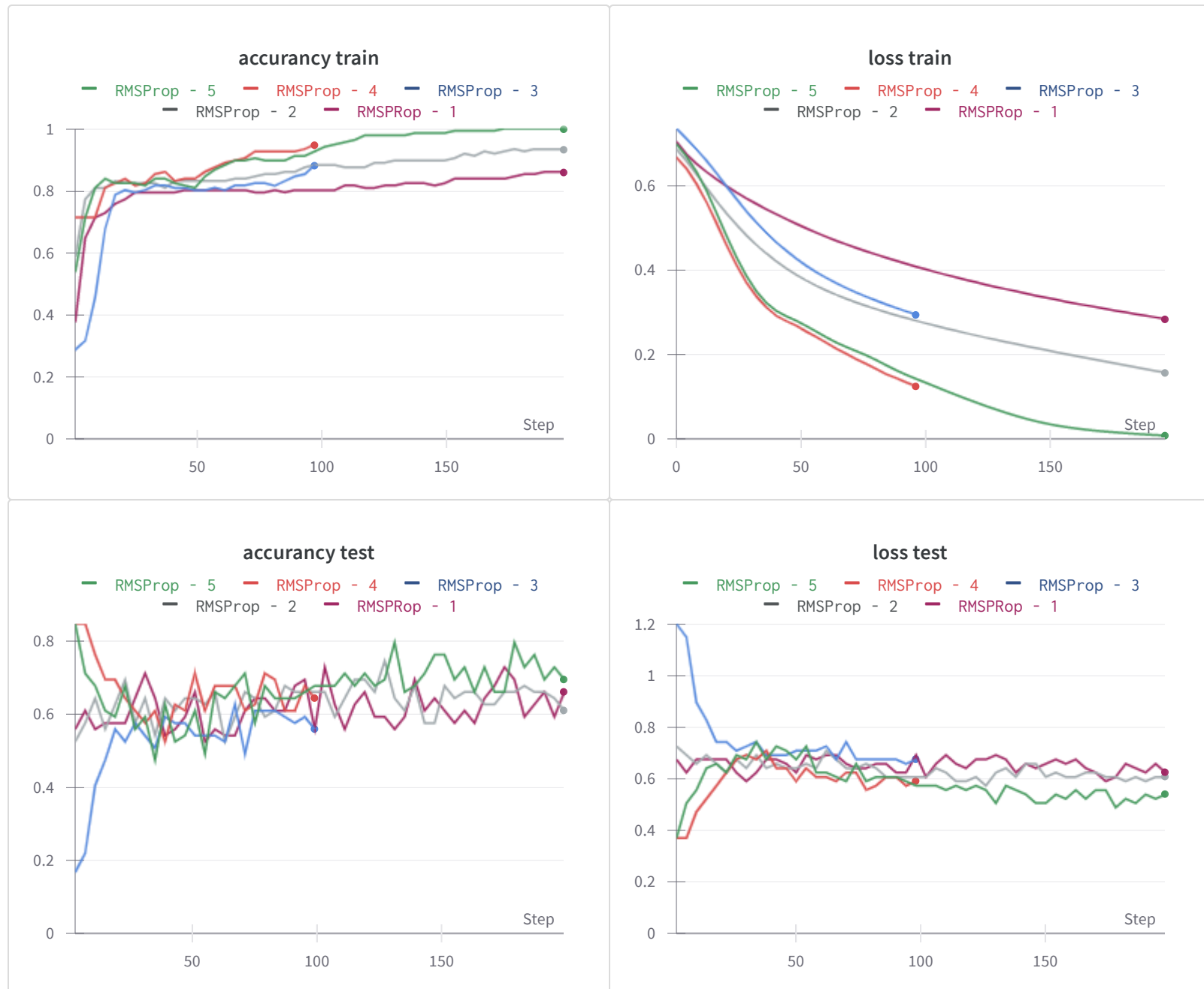
Oba modely si držali približne rovnaké hodnoty, aj keď RMSProp 3 bola nainicializovaná s trochu nižšími váhami. Krivka lossu mala v oboch prípadoch hodnoty veľmi blízko pri sebe.

RMS Prop 4 a 5

Na základe pozitívnych skúseností z predchádzajúcich dvoch modelov sme vyskúšali hodnotu momenta ešte zvýšiť, a to až na 0.9. Opäť sme vytvorili dvojicu modelov, jeden s menším a druhý s väčším počtom epôch. Zvýšenie momenta prinieslo lepšie výsledky ako na testovacích, tak aj na trénovacích dátach oproti hodnote 0.5 v predchádzajúcej dvojici modelov.

Zaujímavým javom v grafoch je podobná hodnota lossu aj accuracy v jednotlivých dvojiciach.

Na porovnaní medzi 2-3 a 4-5 vidíme, že zvýšený počet epôch modelu ešte pomohol, aby sa zvýšili hodnoty accuracy (nebol ešte dotrénovaný).



Záver

Medzi optimizermi RMSProp a Adam sme zaznamenali rozdiel, že Adam mal omnoho lepšie výsledky hneď od začiatku, na rozdiel od RMSProp, ktoré bolo nutné upravovať a vylepšovať pomocou parametrov.

Avšak po upravení parametrov, zvýšení momenta v prípade RMSProp sme dosiahli vyššie hodnoty presnosti na testovacej sade oproti optimizeru Adam.

V porovnaní jednotlivých frameworkov PyTorch / Tensorflow je jasné, že komplexita práce a prípravy dát je v prípade PyTorchu-u omnoho vyššia, nakoľko tento framework je na nižšej úrovni oproti Tensorflow-u.

Created with  on Weights & Biases.

https://wandb.ai/nn2021/parkinson_pytorch/reports/PyTorch--Vmlldzo1Njl3NDM