Diskuze úlohy a řešení

Peter Hroššo pro QMiners

24.2.2019

1 Meta

Podle "bible" přijímacích procesů do technologických společností [2] existují 4 druhy interview otázek / úloh:

- **Sanity check** - kontrola základních schopností
- **Quality check** - (ideálně) sekvence úloh s progresivní úrovní obtížnosti, pomocí kterých se zjistí, kde je horní hranice kandidátových schopností
- **Specialist question** - úlohy testující konkrétní doménovou znalost
- **Proxy question** - úlohy které nesouvisí přímo s danou pozicí, jejichž schopnost řešení ale koreluje s relevantními schopnostmi, které se ale těžko testují přímo. Nejčastějším příkladem jsou datové struktury a algoritmické úlohy

Na základě této taxonomie bych tuto úlohu zařadil někde mezi 1) Sanity check a 3) Specialist question, ale pro jinou specializaci (quantitative analyst), než je vhodná pro pozici, o kterou se ucházím. Pro pozici senior machine learning guru by byla vhodnější spíš úloha typu 2) Quality check a 3) Specialist question, zaměřená na ML / DL. Pokud tedy chci ukázat své kvality, nestačí úlohu pouze vyřešit, musím se pokusit ukázat něco navíc.

2 Zvolený postup

Na základě východisek ze Sekce 1 jsem se rozhodl zvolit následující postup:

- dbát na systematický přístup
- správně prioritizovat
- vyzkoušet větší množství modelů a empiricky vybrat ten nejlepší
- vyzkoušet alespoň jednu neuronovou síť, i když neočekávám příliš dobré výsledky

2.1 Prioritizace

Reálné úlohy jsou málokdy vyřešitelné ideálně. Většinou je tam nějaká forma trade-offu (kvalita vs. čas, ...), takže je potřeba řešit nejdříve ty části, které mají největší očekávaný impakt. V předchozím reportu jsem už zmiňoval známou data science poučku, podle které jsem postupoval:

$$data > model > hyperparametry$$

2.2 Data

Zásadní problém je množství dat. Nízký počet samplů předem vyřazuje všechny složitější a zajímavější modely, protože u nich očekávám velký overfit.

Způsob reprezentace dostupných dat je celkem přímočarý. U finančních dat jsou to spojité hodnoty, u sezónnosti jsou diskrétní (den v týdnu / měsíci / roce), ale myslím že se nedopouštím velké chyby když je reprezentuji jako spojité. Transformaci spojitých hodnot (normalizace / standardizace) beru jako součást modelu.

Jediný zdroj dat, kterým by bylo zajímavé se zabývat z pohledu způsobu reprezentace, je kalendář ekonomických událostí. Tam by se dalo experimentovat s embeddingem událostí a s jejich kombinací s dalšími údaji - jejich očekávaný ekonomický impakt a měna / geografická oblast, které se daná událost týká. Dále by šlo vyzkoušet různé způsoby agregace více událostí jednoho dne.

Ve svém řešení jsem použil jednoduchý one-hot encoding událostí s agregací součtem. Alternativně by šlo brát maximum, minimum, nebo je dát modelu všechny najednou, ať se naučí, které jsou v jakých situacích důležité.

To by samozřejmě přinášelo další komplikace z důvodu proměnlivého počtu událostí v jednotlivých dnech, ale ani to není nic neřešitelného: cut-off +

padding / subsampling při tréninku, a v produkci proměnlivá délka vstupu za použití rekurentních sítí.

Všechny tyto možnosti jsem z důvodu mého časového omezení a jejich nižší priority zavrhl. Kvůli nízkému počtu samplů jsem neočekával, že by tyto složitější reprezentace mohly přinést mnoho užitku.

2.3 Model

Vzhledem k nízkému množství dat jsem předpokládal, že jednodušší modely budou mít lepší výsledky a zároveň mají výhodu dobré interpretovatelnosti. Konkrétně lineární regrese, ideálně s regularizací (Ridge, Lasso, ElasticNet). Chtěl jsem ale vyzkoušel i stromy a support vector regression, což byla ještě relativně nedávno state of the art metoda.

V různých data science soutěžích standardně vyhrávají ensemble metody (random forests, gradient boosting, bagging). Proto jsem je chtěl také vyzkoušet, ale jak se ukázalo, tak na ně už opět nebyl dostatek dat.

2.4 Hyperparametry

Hyperparametry jsou v prioritách úplně nejníže protože je to už spíš finetuning, nečeká se od toho zásadní zlepšení přesnosti modelu. Takže jsem se jimi nezabýval.

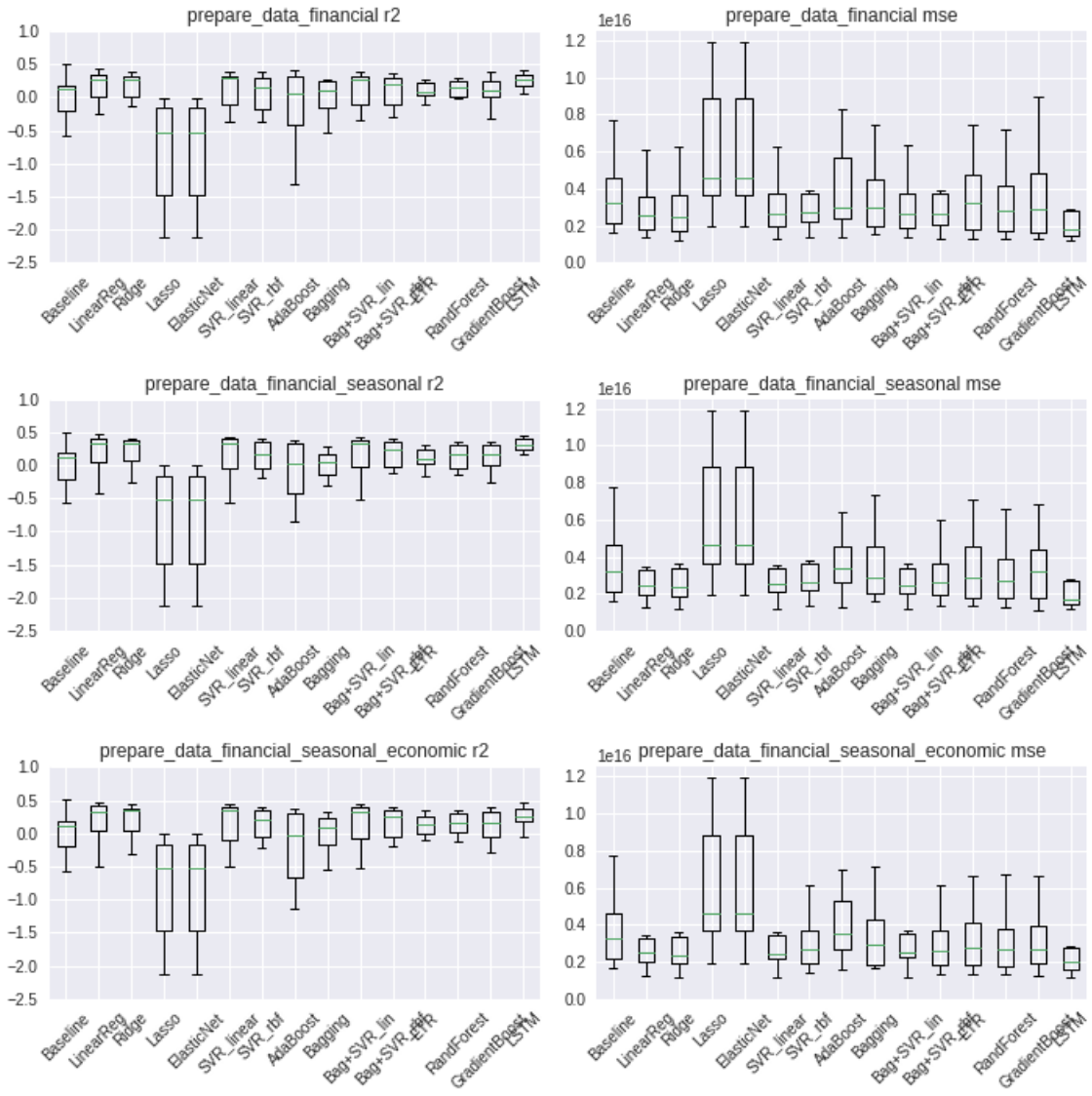
3 Výsledky

Pro úplnost ještě uvádím výsledky finálních experimentů, které mi pro předchozí report nestihly doběhnout. Na Obrázku 1 lze vidět boxplot metrik R2 a MSE pro 10-fold cross-validaci nad 3 druhy přípravy dat: Financial, Financial + Seasonal, Financial + Seasonal + Calendar of economic events.

Nejlépší skóre získala překvapivě neuronová síť LSTM (o jednom neuronu, tzn. nejmenší možná architektura. Experimentoval jsem i s 2-vrstvou, ale už při 2 neuronech silně overfittovala i přes vysokou regularizaci dropoutem, takže jsem ji do závěrečných experimentů ani nedával.) V těsném závěsu je Random Forest (a Extra Trees).

Zajímavé je sledovat efekt přidaných dat. Data o sezónnosti žádnému modelu kromě baggingu a LSTM nepomohla, a pokud ano, tak pouze za cenu většího rozptylu přes jednotlivé běhy cross-validace. Podobný efekt mělo

i přidání dat s událostmi ekonomického kalendáře, a to už i na LSTM a bagging.



Obrázek 1: Boxplot metrik R2 a MSE (levý, respektive pravý sloupec) na 10-fold cross-validaci pro 3 verze přípravy dat: Financial, Financial + Seasonal, Financial + Seasonal + Calendar

3.1 Výsledky na testovací množině

Vybraný model: Jednovrstvá LSTM síť s 1 neuronem, trénovaná na finančních datech a datech o sezónnosti.

Test set	R2	MSE
Baseline	-0.11	1.51E+15
LSTM	0.29	1.10+15

Tabulka 1: Výsledky evaluace nejlepšího modelu vybraného na validační sadě ve srovnání s baseline

4 Možnosti rozšíření

V této sekci nastíním, jak by úloha měla vypadat, aby byl lépe využit potenciál DL, a jak bych v takovém případě přistupoval k jejímu řešení.

4.1 Lepší data

Malé množství dat jsem zmiňoval již několikrát, takže to je jasný kandidát na zlepšení, ale určitě ne jediný. Jde také o rozsah / diverzitu informace v datech obsažených. Z mého pohledu by bylo ideální zkombinovat co nejvíce různých zdrojů. Deep learning umožňuje zpracovávat i velmi bohaté (a přirozeně velmi zašumělé) zdroje dat, jako jsou text a obraz. Šlo by pracovat s business zprávami a jinými relevantními zdroji, ale také třeba sledovat sentiment na sociálních sítích.

Možná nejslibnější by mohlo být sledovat informace relevantní k fundamentu obchodovaných produktů. Při obchodování s akciemi firmy sledovat její PR, hiring, množství marketingu, aktivitu ve veřejném prostoru, konkurenci.

Při obchodování s komoditami by šlo používat satelitní snímky, jejichž cena v poslední době hodně klesla a už toho využívá několik startupů. Sledovat co nejvíce zdrojů informací na kterých je cena daných komodit závislá - věci, které ovlivňují jejich produkci či spotřebu.

4.2 Lepší modely

Bohatší data umožní (a budou vyžadovat) použití lepších, složitějších a větších modelů. Nejdůležitější mi přijde dát modelu dostatečný kontext, s čímž můžou pomoct např. dilated convolutions [3]. Přínosné by také mohlo být zkombinovat různé periody vstupů, např. jeden s denní frekvencí dat, druhý s měsíční, třetí s roční, čímž dáme modelu lepší kontext.

Dalším důležitým prvkem je attention [4]. Ta umožňuje modelu efektivněji pracovat se vstupy a naučit se kauzální vztahy mezi sekvencí vstupů a výstupů bez nutnosti použití rekurentní sítě. Model se naučí která část vstupu je v jakém případě důležitá pro predikci výstupní sekvence. Dále tento přístup umožňuje lepší interpretabilitu modelu, protože naučené kauzální vztahy lze snadno vizualizovat (lze sledovat aktivitu attention a porovnat, na co se síť *"dívá"* při predikci čeho).

4.3 Alternativní formulace úlohy

4.3.1 Auto-regrese

Ve svém řešení jsem vycházel z formulace úlohy jako problém regrese, konkrétně predikce volume, což je nejpřímější přístup vzhledem k zadání. Další možností, jak tuto úlohu řešit, je pomocí autoregrese. Tzn. dívat se na vstupní data jako na stream dat a predikovat jeho další kroky v čase - nejen cílovou proměnnou, ale i všechny ostatní vstupy.

To, že úlohu vlastně ztížíme a *"chceme toho po modelu více"*, má možná trochu kontraintuitivně potenciál zlepšit jeho výsledky. Model je totiž nucen stream dat trackovat věrněji, má méně volnosti a možnosti odchýlit se od reality. To má regularizační účinek a zvyšuje schopnost modelu generalizovat na neznámá data.

Dále nám to opět dává lepší možnosti jak model interpretovat, protože můžeme analyzovat i ostatní predikované proměnné. Např. uvidíme, že ve chvíli, kdy predikuje zásadně špatně volume, dělá chybu i v predikci nějakého konkrétního druhu ekonomické události. To nám napoví, že události nevhodně reprezentujeme nebo daná událost třeba není v datech dostatečně zastoupena.

Případná nevýhoda je, že naše cílová proměnná je při autoregresi *jen jednou z mnoha* a tudíž může být pro model jednodušší ji ignorovat a chybu kompenzovat pomocí přesnější predikce ostatních proměnných. To lze ale

korigovat pomocí custom loss, která bude chybu predikce penalizovat tak, aby to zvýšilo váhu naší cílové proměnné.

4.3.2 Reinforcement learning

Alternativně bychom mohli odhlédnout od regrese a formulovat úlohu v kontextu tzv. reinforcement learningu (RL) jako maximalizaci expected rewardu. Pro trading by to tedy obnášelo definovat si reward funkci, tj. cíl, a akce které model může pro dosažení tohoto cíle používat.

Takovýto způsob formulace úlohy má tu výhodu, že můžeme optimalizovat přímo to, o co nám jde - např. maximalizovat zisk. Navíc se model může naučit pracovat s veškerými detaily, které jsme doposud ignorovali, ale v praxi jsou významné - např. že akce něco stojí, jsou nedokonalé (noisy, zpoždění, ...), můžeme je vykonávat omezeně často, máme omezené zdroje, atd.

Na druhou stranu to opět přináší určité nevýhody - cenou za zabalení celého problému do jedné optimalizační úlohy je snížená interpretabilita modelu.

Z tohoto důvodu by mi připadalo nejlepší přístupy kombinovat - vystavět autoregresní model, který využije maximum informace dostupné v datech v režimu unsupervised learningu, a nad ním potom učit konkrétní akce v režimu RL. Obecně se ukazuje, že nejlépe fungují systémy trénované end-to-end, ale pokud je interpretabilita důležitá a pokud unsupervised learning umožní lépe vytěžit data, pak může tento přístup přinést lepší výsledky.

Napadají mě ještě další směry, kterými by bylo zajímavé se vydat (např. adversarial learning [1]), ale nejdřív je třeba vyzkoušet state of the art a zjistit, co funguje na daný problém, a co ne a proč.

Reference

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. pages 2672–2680, 2014.

[2] G. L. McDowell. *Cracking the coding interview: 189 programming questions and solutions.* 2016.

[3] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.