NSU peta domača naloga
Časovne vrste

Tine Markočič

Avgust 2023

1 Podatki in modeli

Uvozimo datoteko `okuzeni.csv`, in ker nas zanimajo le podatki za mestne občine, pobrišemo vse ostale stolpce. Opazimo, da v občini Sloevnj Gradec manjkajo podatki za 26 dni. Sklepamo, da tiste dni ni bila prijavljena nobena okužba, zato lahko manjkajoče vrednosti nadomestimo z 0.

Karakteristike podatkov namigujejo na to, da jih bo potrebno normalizirati. Ker pa pri napovedovanju ne poznamo vseh podatkov v naprej, izvedemo normalizacijo po sekvencah. V učni zanki modela vsakič pred izvedbo korakov sekvenco normaliziramo:

$$X' := \frac{X - \overline{X}_{sekvence}}{\sigma_{sekvence}}.$$

Nato izračunamo še normalizirano vrednost y s pomočjo povprečja in standardne deviacije prejšnje sekvence. Ker bomo primerjali napovedi s pravimi podatki, moramo dobljene vrednosti števila okužb še odnormirati:

$$y' := \sigma_{sekvence} \cdot y + \overline{X}_{sekvence}.$$

Na podatkih bomo učili nevronske mreže, ki jih sestavljajo celice, ki smo jih spoznali na vajah. Z uporabo knjižnice `Pytorch` implementiramo modele `RNN`, `GRU` in `LSTM`.

2 Učenje in testiranje modelov

Naše podatke sestavlja 800 vrstic, zato bomo vzeli 150 vrstic (približno 20%) za testno množico. Vsako nevronsko mrežo naučimo za napovedovanje za 7 in za 30 dni naprej pri čemer so za vse tri enaki ostali parametri:

- epohi (epochs) = 10
- stopnja učenja (lr) = 0.001
- število skritih plasti (hidden) = 32

- dolžina sekvence (seq_len) = 14

V učni fazi potekajo stvari kot na vajah, saj uporabimo isto for zanko. Do spremembe pride v testni fazi, ko napovedujemo bodisi za 7 bodisi 30 dni v naprej in moramo kodo prilagoditi, tako da sekvenco posodobimo z že napovedanimi dnevi, ki potem služijo za prihodnje napovedi.

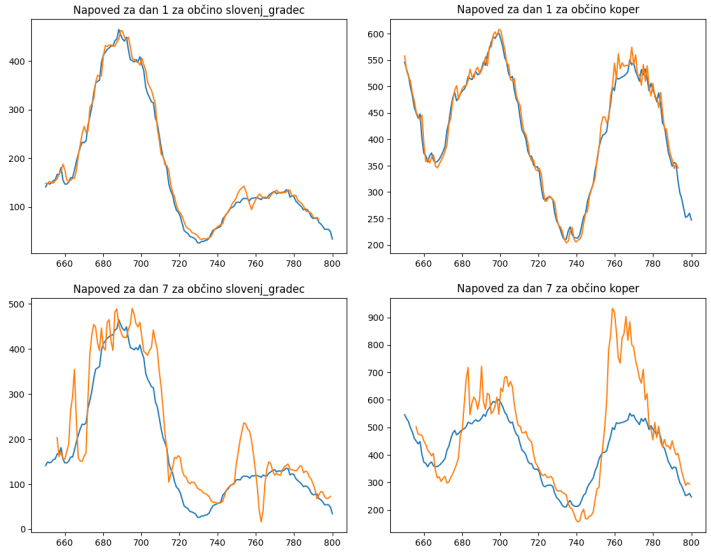
Ko vse modele naučimo, testiramo njihovo natančnost na podlagi napake `R2`. Vse modele združimo v slovar in pogledamo kateri ima v povprečju za izbrano časovno obdobje (7 ali 30 dni) najboljšo natančnost.

3 Analiza napovedi za M=7

Za sedem dnevno napoved je povprečju najboljši model `LSTM` z

$$\text{povp}(R2) = 0.7539283344931637.$$

Najbolje napove podatke za občino Slovenj Gradec, $R2 = 0.914224241212661$, najslabše pa za občino Koper, $R2 = 0.5083667591294087$. Za primerjavo si lahko ogledamo še grafa napovedi za cez 1 dan in za cez 1 teden za obe občini.



Slika 1: Primerjava krajše časovnih napovedi za občini

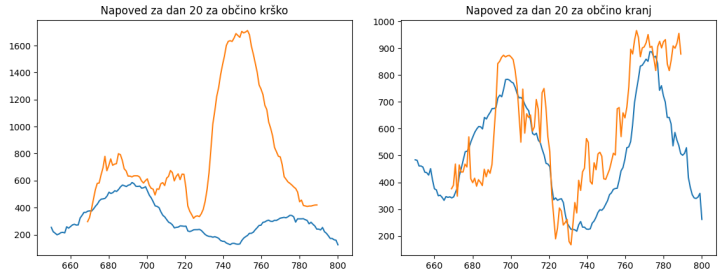
Vidimo, da za napoved okužb v naslednjem dnevu model deluje precej dobro, pri napovedovanju za naslednji teden pa že prihaja do velikih nihanj. Natančnost torej pada ob večanju napovednega obdobja.

4 Analiza napovedi za M=30

Za trideset dnevno napoved je povprečju najboljši model `RNN` z

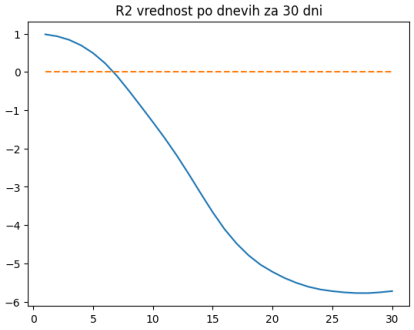
$$\text{povp}(R2) = -3.0804909039881996.$$

Po pričakovanju je povprečna natančnost vseh modelov za to časovno obdobje zelo slaba. Model `RNN` najbolje napove okužbe za občino Kranj, najslabše pa za občino Krško.



Slika 2: Primerjava daljše časovnih napovedi za občini

Vidimo, da model popolnoma zgreši napoved za občino Krško, ko pričakuje ponovno rast okužb v zadnjih 100 dneh. Za občino Kranj sicer izgleda, kot da je zajet približen trend okužb, a se napovedi iz dneva v dan vseeno preveč razlikujejo od resničnih vrednosti. Zaključimo lahko, da naš model ni primeren za napovedovanje okužb za daljše časovni obdobje.



Slika 3: Natančnost modela v odvisnosti od časa

Naš sklep potrdi tudi graf odvisnosti napake od časovne oddaljenosti napovedi modela `RNN`. Model je zares zanesljiv le za napovedovanje v prvih nekaj dneh, ko vrednosti sledijo nekemu trendu, nato pa njegova natančnost preveč pade. Razlog za to so tudi nekateri zunanji dejavniki, ki jih samo iz številk model ne more razbrati. Torej, če želimo boljši model za daljše časovno obdobje, potrebujemo še kakšno dodatno informacijo, recimo povezave med občinami, kar pa je tema napredne naloge.