Rekurzív és egylépéses idősorelőrejelzők pontosságának vizsgálata

Szakos Máté Lichter Bertalan

Kivonat

A gyakorlatban használt előrejelző modellektől az esetek többségében egy előre ismert előrejelzési horizonton szeretnénk tudni az idősorok jövőbeli értékeit. Ez elsősorban abból adódik, hogy a legtöbb alkalmazásban a döntések fix jövőbeli értékekre épülnek, például egy napra, hétre előrejelzett értékekre. Ha megvizsgáljuk a legelterjedtebb idősorelőrejelző modelleket, azt láthatjuk, hogy a legtöbb esetben rekurzív előrejelzést alkalmaznak, azaz a jövőbeli értékeket a múltbeli értékekből, pontonként számítják ki. Ez sokszor jó illeszkedést tesz lehetővé, azonban megjelenhet az úgynevezett drift jelenség. Ennek lényege, hogy a modell kis hibával előrejelzi a jövőbeli értékeket, azonban a hibák összeadódnak, így a jövőbeli előrejelzések pontossága csökkenhet. Ezzel szemben az egylépéses előrejelzés esetén a modell minden egyes jövőbeli értéket külön számít ki, így elkerülhető a drift jelenség. Utóbbi megközelítés hátránya azonban, hogy a legtöbb esetben rosszabb illeszkedést eredményez, mivel a függvény, amelyet megközelít, lényegesen bonyolultabb, mint a rekurzív előrejelzés esetén. Jelen házi feladat célja, hogy a két megközelítés pontosságát összehasonlítsa, és megvizsgálja, hogy melyik esetben érhető el jobb előrejelzési teljesítmény. Házi feladatunkban azt találtuk, hogy stacioner, periodikus idősorok esetén a rekurzív előrejelzés sokszor kevésbé pontos, mint az egylépéses előrejelzés. Az interferáló jel, illetve a fehér zaj mértéke nagy hatással van a pontosságra, azonban szinte minden esetben az egylépéses előrejelzés volt a pontosabb.

Háttér és motiváció

A problémakör egy gyakorlati alkalmazásban merült fel, amiben egy nagyon zajos és komplex nemlineáris, azonban valamelyest periodikus jelet kellett előrejelezni egy viszonylag hosszú időtávra. Intuitíven úgy tűnt, hogy a háttérben álló differenciálegyenletre való visszavezetés lenne a legjobb megoldás, amely kiértékelése megadja a jövőbeli értékeket (ez a rekurzív előrejelzés). Ennek a megközelítésnek az a motivációja, hogy lényegében egy differenciálegyenletet próbál megtanulni a modell, amely generálja a jelet. Ilyen módon nem csak pontos illeszkedést lehet elérni bizonyos esetekben, hanem jól magyarázható modelleket is kapunk [1]. Ez a következő formát öltené:

$$
\begin{aligned}
\dot{x}(t) \approx x(t+\Delta t) - x(t) &= f\left(x(t), x(t-\Delta t), \ldots, x(t-n\cdot\Delta t)\right) \\
&\quad + \epsilon(t+\Delta t) \\
\implies x(t+m\cdot\Delta t) &= x(t) \\
&\quad + \sum_{i=0}^{m-1}\Bigg(f\left(x(t+i\cdot\Delta t), x(t+(i-1)\cdot\Delta t), \ldots, x(t+(i-n)\cdot\Delta t)\right) \\
&\qquad + \epsilon_{\text{rekurzív}}(t+m\cdot\Delta t)\Bigg) \\
&= x(t) \\
&\quad + \sum_{i=0}^{m-1} f\Big(x(t+i\cdot\Delta t), x(t+(i-1)\cdot\Delta t), \ldots, x(t+(i-n)\cdot\Delta t)\Big) \\
&\quad + \sum_{i=0}^{m-1} \epsilon_{\text{rekurzív}}(t+m\cdot\Delta t)
\end{aligned}
$$

Ahogy az jól látszik, hogy a $\epsilon_{\text{rekurzív}}(t+\Delta t)$ hibák akkumulálódnak, így minél távolabbra szeretnénk előrejelezni, annál nagyobb hibával kell számolnunk.

Egy másik megközelítés, amelyet a gyakorlatban gyakran használnak, az egylépéses előrejelzés, amely a következő formát ölti:

$$
x(t+m\cdot\Delta t) = f\left(x(t), x(t-\Delta t), \ldots, x(t-n\cdot\Delta t)\right) + \epsilon_{\text{egylépéses}}(t+m\cdot\Delta t)
$$

Ezen megközelítésnek az az előnye, hogy a hibája nem akkumulálódik, azonban egy nehezebb feladatot old meg, így a hibája általában nagyobb ($\epsilon_{\text{egylépéses}}(t+m\cdot\Delta t) > \epsilon_{\text{rekurzív}}(t+m\cdot\Delta t)$). Ennek ellenére képes pontosabb előrejelzést adni, mivel a rekurzív előrejelzés hibája felhalmozódik, míg az egylépéses előrejelzés hibája nem:

$$
\text{az egylépéses modell jobb} \iff \epsilon_{\text{egylépéses}}(t+m\cdot\Delta t) < \sum_{i=0}^{m-1} \epsilon_{\text{rekurzív}}(t+m\cdot\Delta t)
$$

A vizsgált idősor

A vizsgált idősor egy szint etikus, periodikus jel, amelyet a következő képlettel definiálunk:

$$
\begin{aligned}
x(t) &= \sin(a \cdot t + b) \cdot \sin(c \cdot t) + \rho(t) + \epsilon(t), \\
s.t.\ \rho(t) &= \sum_{i=1}^{N} \nu_i \sin(2\pi \cdot i \cdot f_{\text{base}} \cdot t + \phi_i), \\
\epsilon(t) &\sim \mathcal{N}\left(0, \sigma^2\right)
\end{aligned}
$$

A jel egy nemlineáris, periodikus determinisztikus rész, egy periodikus interferáló zaj, és egy fehér zaj összegéből áll. Az interferáló jelet úgy állítottuk össze, hogy egy adott f_{base} alapfrekvenciát, valamint annak felharmonikusait tartalmazza. Ez jól közelíti a valós életben előforduló periodikus zajokat, például az elektromos hálózatokból származó zajokat. A fehér zaj pedig egy normális eloszlású véletlenszerű zaj, amely az érzékelő hibáját hivatott modellezni.

A periodikus jel a következő paraméterekkel definiálható:

- a: a periodikus jel egyik komponensének frekvenciája
- b: a periodikus jel komponensei közti fáziseltolás
- c: a periodikus jel másik komponensének frekvenciája

Az interferáló zaj a következő paraméterekkel definiálható:

- N: az interferáló zaj komponenseinek száma
- ν_i: az interferáló zaj komponenseinek amplitúdója
- f_{base}: az interferáló zaj komponenseinek alapfrekvenciája
- ϕ_i: az interferáló zaj komponenseinek fáziseltolása

A fehér zaj a következő paraméterekkel definiálható:

- σ: a fehér zaj szórása

Miért szintetikus jel?

A feladat során vizsgált idősor egy általunk konstruált, szintetikus jel, melyet azért választottunk, hogy tetszőlegesen hosszú jelet tudjunk generálni, hogy kedvünk szerint állíthassuk a zajokat, illetve a megfigyelési frekvenciát. Mindemellett talán a legfontosabb, hogy a kiértékelésnél így össze tudjuk hasonlítani a tiszta jellel, így a lehető legpontosabb képet kapva a modellek teljesítményéről. Ezzel szemben egy valós idősor bekorlátozna bennünket, hiszen nem tudnánk tetszőlegesen hosszú jelet generálni, és a zajok is adottak lennének. A szintetikus jel segítségével így nagy szabadságot ad, amellyel a különböző kísérletek során éltünk is.

A tanuló modellek

Mindkét megközelítéshez (rekurzív és egylépéses előrejelzés) egyaránt egy 1d konvolúciós neurális hálózatot használtunk, amely a következőképpen van felépítve:

- **Bemenet**: a bemenet egy n hosszúságú ablak, amely az előző n időpont értékeit tartalmazza.

- **1D konvolúciós réteg**: a bemenetet egy 1d konvolúciós rétegen keresztül továbbítjuk, amely k szűrőt használ, és a kimenet mérete $n - k + 1$ lesz.
- **Aktivációs függvény**: a konvolúciós réteg kimenetét egy tanh aktivációs függvényen keresztül továbbítjuk, amely normalizálja a kimenetet, és segít a nemlineáris kapcsolatok modellezésében.
- **Laposító réteg**: a konvolúciós réteg kimenetét egy laposító rétegen keresztül továbbítjuk, amely a kimenetet egy vektorba alakítja.
- **Teljesen összekötött réteg**: a tanh aktivációs függvény kimenetét egy teljesen összekötött rétegen keresztül továbbítjuk, amely a kimenetet a kívánt méretre alakítja.
- **Kimenet**: a kimenet egy m hosszúságú vektor, amely az előrejelzett értékeket tartalmazza.

Az 1d konvolúciós réteg jól rá tud tanulni a periodikus és nemlineáris kapcsolatokra [2] (trend nélküli idősorok esetén), és képes kezelni a zajos adatokat is, így jól alkalmazható a feladat megoldására.

A megvalósítás az alábbi módon történik[1]:

```
class Conv1DNN(nn.Module):
    def __init__(self, input_size, hidden_size, output_size, kernel_size):
        super(Conv1DNN, self).__init__()
        self.conv1 = nn.Conv1d(1, hidden_size, kernel_size)
        self.tanh = nn.Tanh()
        self.fc = nn.Linear(hidden_size * (input_size - kernel_size + 1), output_size)

    def forward(self, x):
        x = x.unsqueeze(1)
        x = self.conv1(x)
        x = self.tanh(x)
        x = x.view(x.size(0), -1)
        x = self.fc(x)
        return x
```

Hiperparaméterek megválaszása

A kísérletek során korán világossá vált, hogy mindkét megközelítés esetén nagy mértékben függ a modell teljesítménye a hiperparaméterek beállításától. A hiperparaméterek közé tartozik a konvolúciós réteg szűrőinek száma, a tanulási ráta, az ablak mérete, a teljesen összekötött réteg mérete és a tanulási ciklusok (epoch-ok) száma.

Megválasztásuk először pár manuális próbálkozással kezdődött, amellyel felmértük, hogy mely paraméter milyen tartományban működik jól. Ezt követően egy rácskereséses módszert alkalmaztunk, amelyben a hiperparaméterek külön-

[1] A kódot saját implementációként mutatjuk be, de a PyTorch dokumentációja és példái alapján készült. Lásd: PyTorch Conv1d dokumentáció.

böző kombinációit teszteltük, és a legjobb teljesítményt nyújtó kombinációt választottuk ki.

Megvalósítás

A kódot tartalmazó jegyzetfüzetet itt lehet elérni. A megvalósítás során erősen építettünk a `pytorch` könyvtárra, amely lehetővé teszi a neurális hálózatok egyszerű és hatékony megvalósítását. A kiértékeléshez a `permetrics` könyvtárat használtuk, amely segít az idősorok előrejelzésének kiértékelésében. A kirajzoláshoz pedig a `matplotlib` és `seaborn` könyvtárakat használtuk, amelyek lehetővé teszik a vizualizációk egyszerű és szép elkészítését.

Kísérletek

A kísérletek során minden esetben alkalmazva voltak a következő paraméterek:

Paraméter	Érték
a	0.12123123
b	2.132123
c	5
σ	0.1
f_{base}	5
f_s	2

Utóbbi (f_s) a megfigyelési frekvencia, amely a jel mintavételezésének gyakoriságát határozza meg.

A generált jel 700 másodperc hosszú, amelyből az első 40%-ot tanulásra, 10%-ot validációra, és a maradék 50%-ot tesztelésre használtuk. Fontos megjegyezni, hogy mindenhol a zajos jelet használtuk, kivéve a tesztelésnél a kiértékeléshez; ezzel mimikálva a valós életben előforduló helyzeteket (ahol nem áll rendelkezésre a tiszta jel, azonban a modell helyességét a tiszta jelhez viszonyított hibája határozza meg).

Az eredmények kiértékelése

Az egyes modellek tanítását (hiperparaéterek optimalizációjával) követően a teljes teszt adathalmazon kiértékeltük (minden olyan ablakon, amely a visszatekintési ablakot figyelembe véve belefér a tesztelési adathalmazba) a modellek teljesítményét. Minden modell minden előrejezési horizonton ki volt értékelve három pontossági metrika szerint is.

fig:baseline_histograms

1. ábra: fig:baseline_histograms

A pontossági metrikák

Determinisztikus együttható (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Átlagos Arkusztangens Abszolút Százalékos Hiba (MAAPE)

$$\text{MAAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\arctan(y_i) - \arctan(\hat{y}_i)}{\arctan(y_i)}\right| \cdot 100$$

Normalizált Gyökös Középérték Négyzetes Hiba (NRMSE)

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{\max(y) - \min(y)}$$

Alapeset (baseline)

A vizsgált jel paraméterei:

Paraméter	Érték
N	8
ν_i	e^{-i}
ϕ_i	0

Eredmények TODO: eredmények beszúrása

TODO: másik 3 eset bemutatása

Konklúzió

A kísérletek során sikerült bemutatni, hogy a rekurzív és egylépéses előrejelzés is képes jól teljesíteni a vizsgált idősor előrejelzésében, azonban legtöbb esetben az egylépéses előrejelzés volt a jobb teljesítményű (különösen a hosszú távú előrejelzésnél). A rekurzív előrejelzés esetén a hibák felhalmozódása miatt a hosszú távú előrejelzés pontossága csökkent, míg az egylépéses előrejelzés esetén a hibák nem halmozódtak fel, így a hosszú távú előrejelzés is pontosabb volt.

Limitációk

A házi feladat során csupán egy architektúrát vizsgáltunk, egyetlen idősoron. Előfordulhat, hogy más architektúrák vagy idősorok esetén más eredmények születnének. Elképzelhetőek olyan modellek is, amelyek nem tisztán az egyik vagy más megközelítést használják, hanem egyesítik a rekurzív és egylépéses előrejelzést, így kihasználva mindkét megközelítés előnyeit [3].

Irodalomgyűjtemény

[1] K. Egan, W. Li, és R. Carvalho, „Automatically discovering ordinary differential equations from data with sparse regression", *Communications Physics*, köt. 7, sz. 1, jan. 2024, doi: 10.1038/s42005-023-01516-2.

[2] S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci, és M. Gabbouj, „1-D Convolutional Neural Networks for Signal Processing Applications", in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, o. 8360–8364. doi: 10.1109/ICASSP.2019.8682194.

[3] S. B. Taieb, R. J. Hyndman, *és mtsai.*, *Recursive and direct multi-step forecasting: the best of both worlds*, köt. 19. Department of Econometrics and Business Statistics, Monash Univ., 2012.