

FlexCoder:

Gyakorlati programszintézis flexibilis input-hosszokkal és kifejező lambdafüggvényekkel

Mucsányi Bálint Gyarmathy Bálint Czapp Ádám
Témavezetők: Dr. Pintér Balázs Dr. Gregorics Tibor

Országos Tudományos Diákköri Konferencia
Eötvös Loránd Tudományegyetem, Informatikai Kar

2021 március 31.



A szintézis folyamata

Be: [1, 5, 4, -2, 6]
Ki: [2, 10, 8]



Szintetizálási
folyamat



```
map(*2, take(3, array))
```

Függvénykompozíció

```
[1, 5, 4, -2, 6]
```



```
map(*2, take(3, array))
```

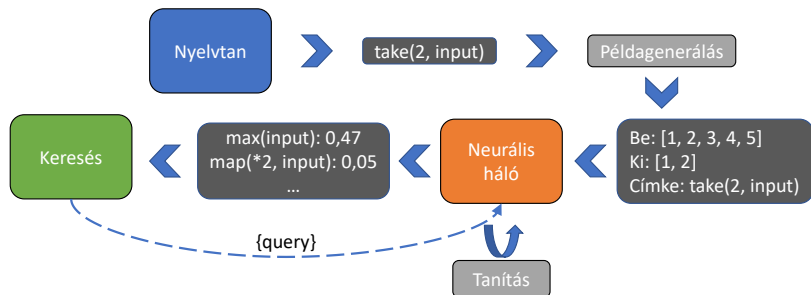


```
[2, 10, 8]
```



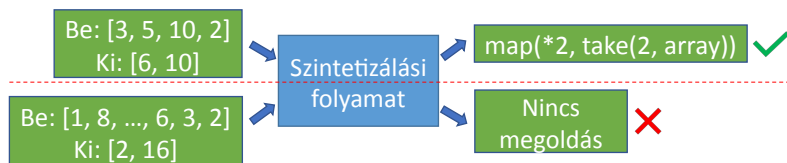
Feladat, komponensek

- Feladat
 - Programok szintetizálása input-output párok alapján
- A neurális programszintézis-rendszerek főbb elemei:
 - Nyelvtan és adatgenerálás
 - Mélytanuló algoritmus \Rightarrow nagy mennyiségű adat szükséges
 - Neurális háló
 - Keresőalgoritmus
- Hogyan működnek együtt a komponensek?



Főbb limitációi a legjobb rendszereknek:

- Statikus vagy felülről korlátos listaméret



- Nyelvtani elemek agglutinálása
 - `map(*2, list)`
- Lambdaoperátorok limitált paramétertartománya
- Az inputok limitált tartománya

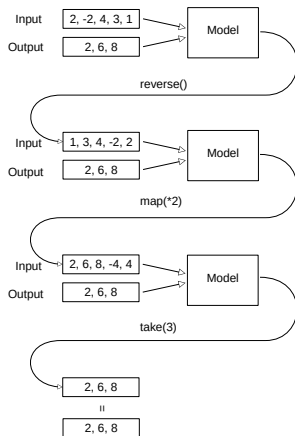


A nyelvten leegyszerűsített változata

$S \rightarrow \text{ARRAY_FUNCTION} \mid \text{NUMERIC_FUNCTION}$
 $\text{ARRAY_FUNCTION} \rightarrow \text{take}(\text{NUM}, \text{ARRAY}) \mid \text{drop}(\text{NUM}, \text{ARRAY})$
 $\text{ARRAY_FUNCTION} \rightarrow \text{map}(\text{LAMBDA_FUNCTION}, \text{ARRAY})$
 $\text{ARRAY_FUNCTION} \rightarrow \text{filter}(\text{LAMBDA_FUNCTION}, \text{ARRAY}) \mid \dots$
 $\text{NUMERIC_FUNCTION} \rightarrow \text{max}(\text{ARRAY}) \mid \text{min}(\text{ARRAY})$
 $\text{NUMERIC_FUNCTION} \rightarrow \text{sum}(\text{ARRAY}) \mid \text{count}(\text{ARRAY}) \mid \dots$
 $\text{NUM} \rightarrow -8 \mid \dots \mid 8$
 $\text{LAMBDA_FUNCTION} \rightarrow \text{OPERATOR NUM}$
 $\text{OPERATOR} \rightarrow * \mid / \mid + \mid \% \mid > \mid == \mid \dots$
 $\text{ARRAY} \rightarrow \text{list} \mid \text{ARRAY_FUNCTION}$

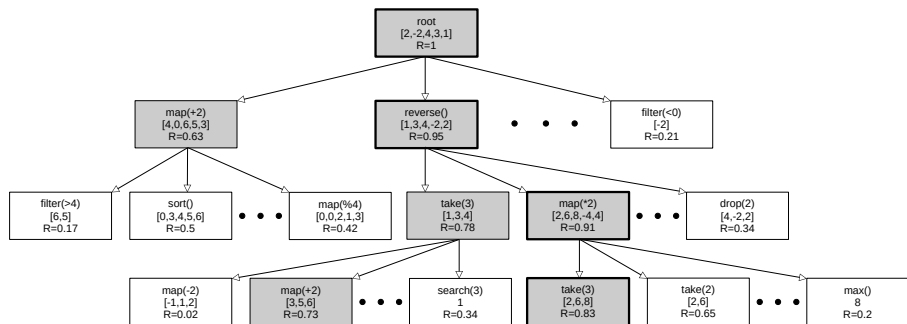
$S \rightarrow \text{NUMERIC_FUNCTION} \rightarrow \text{max}(\text{ARRAY}) \rightarrow \text{max}(\text{ARRAY_FUNCTION})$
 $\rightarrow \text{max}(\text{map}(\text{LAMBDA_FUNCTION}, \text{ARRAY})) \rightarrow \text{max}(\text{map}(*2, \text{list}))$





- Az input(ok) és a hozzájuk tartozó output(ok) adottak
- A kompozíciót egyszerre egy függvénnyel bővítjük
- A neurális háló heurisztikaként irányítja a keresést
- `take(3,map(*2,reverse(inp)))`

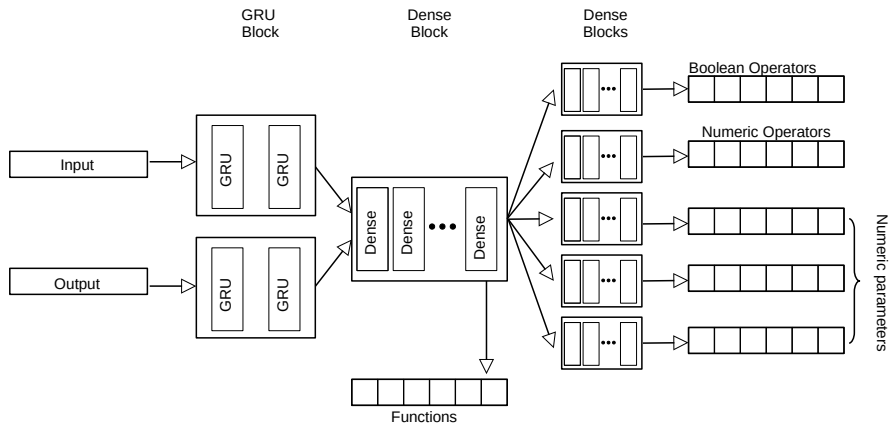


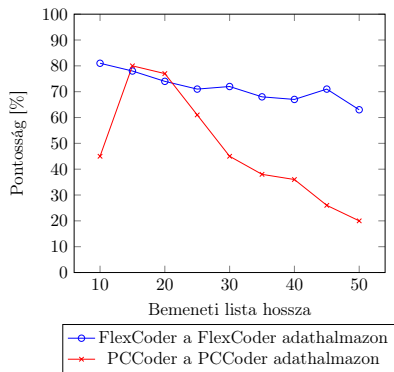
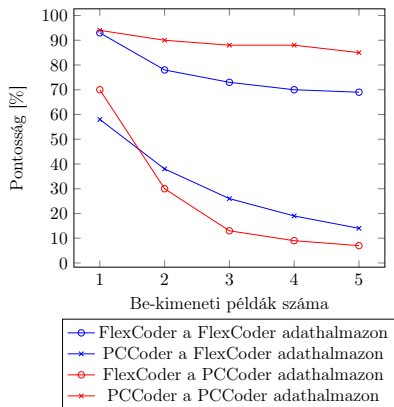


- A sikerre vezető ág fekete kerettel van jelölve



A neurális háló strukturája





- A mi rendszerünk: FlexCoder¹
- A Zohar, L Wolf: PCCoder²

¹Gyarmathy, B., Mucsányi, B., Czapp, A., Szilágyi, D., & Pintér, B. (2021). Flexcoder: Practical Program Synthesis with Flexible Input Lengths and Expressive Lambda Functions.

²Zohar, A., & Wolf, L. (2018). Automatic program synthesis of long programs with a learned garbage collector.



- Egy programszintetizáló rendszert mutattunk be, ami jól általánosít különböző bemeneti hosszokra.
- A lambdafüggvények operátorait külön kezeljük azok paramétereitől.
- A köztes állapotok és kimenetek tartományát négyszeresére növeltük az előző munkákhoz képest.

Publikáció a 2021-es *International Conference on Pattern Recognition Application and Methods (ICPRAM)* konferencián

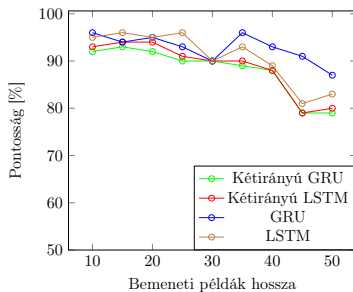
- Jelölés a konferencia legjobb cikkének járó díjra
- Felkérés egy bővített cikk írására a *Springer Nature Computer Science Journal* kiadványba



- Ha a háló függvényt ad kimenetként, hol alkalmazzuk a nyálábkeresést, illetőleg ha nem azt, (hanem pl. a rangot), akkor minden predikció után végre kell azt hajtani?
- Hogyan választották ki a rekurrens háló szerkezetét? Miért pont 7 réteg és miért pont annyi a neuronszám az egyes rétegekben?
- Hogyan történt az inputpéldák generálása? Minden generált példa esetén létezett legalább egy valid kompozíció, amely a bemutatott CFG-vel definiálható? Azaz a módszer elméletileg minden példához találhatott megoldást?
- Mi volt a pontosság metrika, hogyan mérték a teljesítményt (accuracy, precision, más)?



- Melyik platformot használták? TensorFlow? PyTorch? Más?
- A 4.3. ábrán (30. oldal) látjuk a pontosság alakulását az inputhossz függvényében több RNN esetén. 30-as inputhosszra mind a 4 RNN pontosan azonos pontosságot adott. Lehet ennek valami technikai oka, vagy pusztán véletlen egybeesés?



- Kérem magyarázzák meg, hogy miért ilyen nagy százalékát (98%, 26.o) használták a mintáknak a tanításhoz, s miért ilyen kis mintán (2%) teszteltek. Illetve alkalmaztak-e valamilyen keresztvalidálást a kiértékelés során?
- A tanítás során hogyan bizonyosodtak meg arról, hogy a háló megfelelően tanult, s nem áll fenn a túltanulás veszélye?
- Hogyan osztották 6 részre a teljesen összekapcsolt háló kimenetét? Mi volt ennek a célja és mi indokolta ezt?
- Véleményük szerint a későbbiekben milyen gyakorlati haszna lehet a javasolt modellnek?



Köszönjük a figyelmet!

Efop-3.6.3-Vekop-16-2017-00001: Tehetséggondozás és kutatói utánpótlás fejlesztése autonóm járműirányítási technológiák területén – A projekt a Magyar Állam és az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

