Farkas Dániel Szakmai házifeladat:

Második feladat (elméleti):

Szeretnénk neurális hálózatot készíteni, amely képes prediktálni a BTC holnapi árfolyamát. Kétféle képpen állunk neki:

- Először az árfolyam konkrét értékét próbáljuk tanítani és becsülni.
- Másodszor az előző napi értékhez képesti eltérést próbáljuk meghatározni (+/- hány BTC)

Hogyan állna neki a feladat elkészítésének? Milyen eredményeket várna a két esetben? (**A feladatot nem kell megcsinálni**)

A feladat kimenete:

- Szöveges elemzés a feladat megvalósításának lehetséges menetéről, kihívásairól. Tartalmazzon egy becslést is: mennyi idő alatt lehetne a feltételezéseket gyakorlatban is igazolni? Ez utóbbihoz készítsen egy munkaóra becslést az egyes lépésekhez.

Első gondolatok:

A bitcoin árfolyam változása emberi mértékkel nem leírható függvény alapján történik. Egy egyszerű lineáris neurális háló nem képes a predikciójára, azonban több irányból megközelíthető, hogyan lehet mégis betanítani rá egy mesterséges inteligenciát.

Egyes számú megoldás (konkrét adatok tanítása):

Nehézségek:

Ennél a megoldásnál mivel a konkrét értékek alapján szeretnénk a következőt jósolni, ezért olyan hálóra van szükségünk, ami szekvenciális adatfeldolgozást tesz lehetővé. Célszerű ilyenkor visszamutató architektúrát használni.
Ilyen például a gyakorlati házimban használt LSTM (Long short-term memory) modell amit kifejezetten szekvenciális adatokhoz ajánlanak. Mivel a visszamutató hálók jelentősen lassabban tanulnak, ezért még nehezebb meghatározni a rendszer paramétereit. Célszerű több egymást követő számot bemenetnek választani.

Megvalósítás:

Először meg kellene határozni az alap architektúrát, hány file, és milyen leosztásban legyen. Nem kapcsolódik szorosan, de ide venném az inputok és outputok meghatározását, és az adatfeldolgozás menetének kidolgozását is.

Ezt nagyjából 4-8 órára tippelem. (teszteléssel együtt)

Aztán meg kellene határozni pontosan a használt modellt, és megírni a hozzá tartozó segédfüggvényeket is.

Ezt nagyjából 12-16 órára tippelem

Aztán módot kell találni a tanulás pontosságának visszajelzésére is. Ez hasznos lehet teszteknél is, és természetesen a kapott eredmény értékelésére is.

Ezt nagyjából 8 órára tippelem.

Aztán a modell finomhangolására, és tesztelésére is szánnék időt, erre jobb esetben elég lehet 8 óra is, de a worst-case szenárió több napot is elvehet, így:

Ezt nagyjából 8-16 órára tippelem.

Végül a rendszer összeillesztésére, és a környezet (például user interakciók) kidolgozására és tesztelésére is kellene valószínűleg idő, ez nyilván projekt függően teljesen más számokat jelent, vegyünk most egy egyszerű diagrammos megjelenítést, és egy számsor output-ot.

Ezt nagyjából 4 órának tippelem (teszteléssel együtt)

Előnyök:

Nagy előnye az első opciónak, hogy általánosabb megközelítés, így olyan összefüggéseket is megtalálhat, amiket a kettes számú megoldásunk nem. (Ha például karácsonyi csodaként minden decemberben megemelkedik az árfolyam, ezt a periodikusságot ez a módszer észreveheti.) Konkrét értékekkel általánosan bármilyen számsorról szerezhetünk valamennyi információt, delták számolásával azonban nem feltétlen. Hozzá tartozik, hogy ha elég sok adatot dolgozunk fel egyszerre, beleeröltethető a példaként hozott információ többlet a másik módszerbe is, azonban ehhez előzetes információk kellenek, és nem minden összefüggésnél bővíthető a másik rendszer.
Várhatóan pontosabb eredmény.

Hátrányok:

Hosszabb tanítási idő, bonyolultabb neurális háló.

Kettes számú megoldás (előző napokhoz mért delták tanítása):

Nehézségek:

Ennél a megoldásnál egyszerűbb többrétegű hálók is használhatóak lehetnek, hiszen tudjuk, hogy a változás mértéke, és a következő érték között egyenes összefüggés van. Természetesen vannak így nem tanulható külső tényezők, de általában jól lehet tippelni az előző napok elváltozásaiból, a következőre. Komoly veszély lehet, hogy ha csak deltákból számolunk, hogy az egyes kerekítésekből, scale-elésekből, és hasonló megoldásokból származó hibák hasszú távon összeadódhatnak, és teljesen elszállhatnak az adataink. Ennek elkerülésére különböző módok vannak, de figyelni kell rá.

Megvalósítás:

Mint az előző esetben is, először meg kellene határozni az alap architektúrát, hány file, és milyen leosztásban legyen. Nem kapcsolódik szorosan, de ide venném az inputok és outputok meghatározását, és az adatfeldolgozás menetének kidolgozását is.

Ezt nagyjából 4-8 órára tippelem. (teszteléssel együtt)

Aztán meg kellene határozni pontosan a használt modellt, és megírni a hozzá tartozó segédfüggvényeket is.

Ezt nagyjából 8 órára tippelem

Aztán itt is szükség van a tanulás eredményeinek visszajelzésére, ezt is le kell fejleszteni.

Ezt nagyjából 8 órára tippelem.

Aztán a modell finomhangolására, és tesztelésére is szánnék időt, erre jobb esetben elég lehet 4 óra is, de a worst-case szenárió ebben az esetben is problémás.

Ezt nagyjából 4-8 órára tippelem.

Végül a rendszer összeillesztésére, és a környezet (például user interakciók) kidolgozására és tesztelésére is kellene valószínűleg idő, ez nyilván projekt függően teljesen más számokat jelent, vegyünk most egy egyszerű diagrammos megjelenítést, és egy számsor output-ot.

Ezt nagyjából 4 órának tippelem (teszteléssel együtt)

Előnyök:

Legnagyobb előnye a gyorsabb tanulás, valamint az egyszerűbb háló, és nagyobb tanulási sebességből származó gyorsabb debug-olás.

Hátrányok:

Pontatlanabb, kötötteb rendszer.