Neural network voor het voorspellen van de toekomstige prijs van ether

René Steeman

|  |  |
| --- | --- |
| Inleiding | 1 |
| Vraag en antwoord | 1-2 |
| Geschiedenis en ontwikkelingen | 2-4 |
| Werking | 4-5 |
| Precisie en mogelijke verbeteringen | 5-7 |
| Ethica | 7 |
| Uitleg begrippen | 7 |
| Logboek | 8-9 |
| PvA | 9-10 |
| Moeilijkheden | 10 |
| Eindadvies | 10 |
| Tips voor overname | 10 |
| Uitleg/handleiding | 10 |

Inleiding

Ik heb dit onderwerp gekozen omdat ik me af vroeg hoe neural networks en digitale munten werken. De beste manier om hier meer over te leren leek mij het uitvoeren van een project dat iets van beiden heeft. Het resultaat hiervan is mijn proefwerkstuk dat een neural network is dat de toekomstige prijs van ether, de munt van ethereum kan voorspellen.

Vragen en antwoorden

De onderzoeksvraag luidt: Is de prijs van ether te voorspellen met behulp van een neural network en zo ja, hoe goed?

Hyptohese: Ik verwacht dat het mogelijk is om de prijs voor de korte termijn te voorspellen, maar niet voor erg lang aangezien de markt erg onvoorspelbaar is en soms erg onlogisch.

Deelvragen:

1. Wat is de beste structuur voor het neural network?
2. Hoe precies zijn de voorspellingen?
3. Voor hoeveel dagen kan ik redelijkerwijs vooruit voorspellen?
4. Kan ik de waarde voor de volgende dag binnen een minuut voorspellen?
5. Kan ik het neural network op een gpu laten werken?

Antwoorden op de deelvragen:

1. Een recurrent neural network met long short-term memory
2. Zie het hoofdstuk over precisie
3. 1-2 dagen is goed mogelijk
4. Ja, het kan in 6s (inclusief het laden van het model) op een gtx1070 met het model op een hdd
5. Ja, met CuDNN en pytorch voor cuda 9

Wat heb ik moeten doen als voorbereiding?

Het leren van: python, machine learning en deep learning. Het maken van een programma dat de informatie voor het neural network kan verzamelen (ook in python) en kan opslaan als .csv.

Wat heb ik gebruikt voor dit project?

Udemy als platform om op te leren over python, machine learning en deep learning.

Tensorflow en keras voor deep learning.

Spyder als IDE (integrated development environment)

Atom als tekstbewerker waarmee ik het programma heb geschreven dat data ophaalt voor het neural network.

Een korte geschiedenis van geld

Geld is een ruilmiddel om iets met waarde te wisselen voor goederen. In de loop van de tijd is er veel aan veranderd, een kort overzicht van wat als geld gebruikt is/wordt/gaat worden zal volgen.

Voorwerpen/voedsel, zoals: vlees, melk, brood en hout

Metaal, zoals stukken goud en zilver

Muntgeld, kleine stukjes metaal geld met hun waarde erop gedrukt

Papiergeld, stukken papier of stof met hun waarde erop gedrukt

Cheques, papieren belofte dat om te ruilen valt voor papier- of muntgeld

Digitaal geld, een getal in een database dat om te ruilen valt voor fysiek geld

Crypto, nieuwe digitale valuta dat geen fysieke versie heeft en vaak gedecentraliseerd is

De overgang tussen deze valuta ging gepaard met een steeds centralere plek om geld op te slaan. Voedsel werd nog door mensen zelf gehouden, metaal geld werd al gedeeltelijk bij een bank opgeslagen en digitaal geld staat vaak ook nog bij een bank geregistreerd. Bij crypto verdwijnt deze centrale identiteit. Dit digitaal geld kan ‘verplaatst’ worden zonder centrale identiteit, het kan direct doorgestuurd worden, al gaat deze transactie eerst door een ‘controle laag’ om de transactie te controleren en valideren, deze laag bestaat uit mensen die (bij proof of work) de rekenkracht van hun apparaten hiervoor beschikbaar stellen.

Geld is ook steeds meer afhankelijk geworden van vertrouwen, voedsel is nog direct nuttig, maar een cijfer in een database niet, het heeft alleen de waarde van vertrouwen om het ooit om te kunnen zetten in iets dat wel direct nuttig is. Geld is dus ook steeds abstracter geworden, maar heeft nog een paar kenmerken behouden:

Het is moeilijk te vervalsen

Het heeft schaarste (en daarmee waarde)

Het is ‘algemeen’ bruikbaar, oftewel het is om te wisselen voor iets anders

Dit laatste is bij crypto nog erg uitgewerkt, maar daar wordt hard aan gewerkt en er is ook al veel progressie. De snelle veranderingen in waarde en bij sommige munten de tijd die nodig is voor het bevestigen van een transactie houden dit momenteel nog tegen.

Ontwikkelingen AI

Een overzicht van AI-ontwikkeling in chronologische volgorde:

Simple ai voor eenvoudig schaken

AI voor robots in fabrieken

AI voor een chatbot met natural language processing

Zelfrijdende robot

Goede AI voor schaken

Emotionele AI

Deep learning begint met letterherkenning

Zelfrijdende auto (met gebruik van deep learning)

GPU’s voor deep learning

AI als assistenten op een telefoon

Go AI

Waardoor wordt AI (snel) beter?

De kracht van computers neemt nog steeds snel toe, misschien niet meer volgens de wet van Moore, maar alsnog op een goed tempo. Verder is er ook steeds meer data beschikbaar om bijvoorbeeld een AI mee te trainen of om speciale programma’s voor te ontwikkelen (bijvoorbeeld bij de beurs). De groei van AI is exponentieel door deze factoren, verder zijn er ook steeds meer mensen met interesse hierin en is het makkelijker om erover te leren en zelf mee te werken. Dit komt mede door Nvidia die veel aandacht geeft aan het versnellen van AI met GPU’s en software om hier makkelijk gebruik van te maken, daarnaast is er ook steeds meer documentatie en uitleg online te vinden. Zo heb ik veel kunnen leren met behulp van een udemy cursus van maar €12 en gratis software (Keras, CuDNN, Anaconda).

De toekomst:

Ik verwacht dat over 10 jaar een groot deel van banen uitgevoerd kunnen worden door AI, maar dat veel mensen het toch blijven doen. Dit komt simpelweg doordat er geen ander werk voor ze is en ze toch iets te doen moeten hebben en geld moeten verdienen. Op de lange termijn betekent dit dat er een basis inkomen nodig zal zijn, omdat er gewoonweg geen geld meer te verdienen valt met werken voor het grootste deel van de bevolking. Voor deze mensen moet ook iets zinnigs te doen zijn, zodat ze zich niet nutteloos voelen en dit kan nog een van de grootste uitdagingen van de mensheid worden. Net zoals het in bedwang houden van AI, als een AI te slim wordt en teveel kan doen kan dit het einde van de mensheid beteken. Door al deze factoren zou het best kunnen dat de mens in de toekomst half computer is, waar een computer helpt de mens slimmer te maken en zich nog nuttig te laten voelen.

Werking ether voorspellende RNN (simpele versie)

Deel 1, ophalen data over ether:

Ik haal de punten van de grafiek op van coinmarketcap en zet het om in een tabel met daarin alle nodige waarden (datum en prijs worden door de RNN gebruikt, de rest (volume, btc price, high, lows) zouden door een RNN gebruikt kunnen worden voor meer precisie, maar zijn niet in de huidige versie van de RNN geïmplementeerd, aangezien dit het programma veel ingewikkelder zou maken en mij dit geen goed idee leek voor mijn eerste project met AI en python).

Deel 2, het klaarmaken van de data:

Alle waarden die door deel 1 zijn opgehaald worden opgeslagen in een .csv (een soort excel overzicht). Deze worden vervolgens in een array (lijst met waarden) opgeslagen voor de RNN en de datastructuur wordt klaargemaakt zodat de RNN het makkelijk kan gebruiken.

Deel 3, de RNN zelf:

De structuur van deep learning die ik voor dit programma hebt gebruikt heet een RNN, een recurrent neural network. Dit probeert de hersen van een mens na te bootsen door neuronen met elkaar te laten communiceren en gewichten en activeringswaarden aan deze neuronen te binden. De gewichten bepalen voor hoeveel een waarde meetelt. De activeringswaarde geeft aan wat de neuronen die ervoor kwamen als waarden door moeten geven voordat dit neuron een waarde doorgeeft. Uiteindelijk volgt hier een waarde uit.

Een RNN heeft al eigenschap dat neuron hogerop kunnen communiceren met neuronen onderop de hiërarchie. Het bijzondere aan mijn RNN is dat het LSTM heeft, oftewel geheugen waarin vorige waarden opgeslagen kunnen worden. Hierdoor kan het patronen in de resultaten beter herkennen en trends voorspellen.

Het programma kijkt eerst of er al een getraind model is, zo ja, dan laadt hij dit model, zo nee dat traint hij een model door steeds waarden in te voeren en de antwoorden zo goed mogelijk te krijgen (de waarden van ether uit het verleden kunnen hier als antwoorden gebruikt worden) en slaat het vervolgens op in een .h5 en .json bestand (maar 16kb in totaal). In dit model staan de gewichten en activeringswaarden samen met de structuur van de neuronen en manieren waarop waardes overgebracht worden (functies waardoor de waarden tussendoor heen gaan voor betere resultaten).

Deel 4, de RNN gebruiken:

De laatst bekende waarden van de afgelopen dagen worden in de RNN ingevoerd (in een speciale structuur waardoor dit begrijpbaar is voor de RNN), de RNN zal dan de waarde die eruit is gekomen opslaan een nieuwe array met voorspelde waarden. Als er voor meer dan een dag vooruit wordt verspeld worden de voorspelden waarden ingevuld alsof het oude waarden zijn en wordt er weer een antwoord gegeven door de RNN, dit herhaalt zich voor het aantal dagen dat vooruit voorspeld wordt. Er wordt dus steeds met een kleine afwijking verder gewerkt en zal dus steeds minder precies worden.

Deel 5, de resultaten weergeven:

De resultaten worden in een grafiek ingevuld waar de waarden t.o.v. de tijd in de toekomst staan en hieruit kunnen de waarde dus makkelijk door een mens afgelezen worden.

Precisie

Nodige info voor het begrijpen van de tabellen

De waardes worden voor het eind van de dag voorspeld.

Alle $ waarden worden afgerond op gehele getallen.

Afwijking $ = 1 - prijs werkelijk / prijs verwacht (dichter bij 0 is beter)

Afwijking % = Δ$ werkelijk / Δ$ verwacht (dichter bij 1 is beter)

De RNN is voor het laatst getraind (‘nieuwe hersenen gekregen’) op 31-10 en heeft vervolgens niet meer geleerd, alleen waarde gekregen om mee te voorspellen, het heeft dus niet opnieuw verbanden gezocht en beoordeeld.

1 dag vooruit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Datum** | **Oude waarde** | **Verwacht** | **Werkelijk** | **Afwijking $** | **Afwijking % verandering** | **Richting (up/down)** |
| **31-10** | 291 | 304 | 297 | 0.02 | 0.5 | Correct |
| **1-11** | 306 | 299 | 295 | 0.01 | 1.6 | Correct |
| **5-11** | 300 | 299 | 296 | 0.01 | 0.3 | Correct |
| **8-11** | 293 | 298 | 309 | -0.04 | 3.2 | Correct |
| **9-11** | 308 | 316 | 321 | -0.02 | 1.6 | Correct |
| **gemiddeld** | 300 | 303 | 298 | -0.02 | 1.4 | Correct |

Bij een model dat aangepast is voor deze tijdsperiode geldt verwacht nieuw = verwacht + (verwacht-oude waarde)\*1.4 een beter (gemiddeld) resultaat geven, aangezien de markt hier (gemiddeld) minder rustig was dan gebruikelijk. De tabel zou er dan als volgt uit hebben gezien:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum** | **Oude waarde** | **Verwacht nieuw** | **Werkelijk** | **Afwijking % verandering** |
| **31-10** | 291 | 322 | 297 | 0.2 |
| **1-11** | 306 | 289 | 295 | 0.6 |
| **5-11** | 300 | 298 | 296 | 2 |
| **8-11** | 293 | 305 | 309 | 1.3 |
| **9-11** | 308 | 327 | 321 | 0.7 |
| **gemiddeld** | 300 | 303 | 298 | 1.0 |

Dit komt voornamelijk doordat 8-11 een plotselinge verandering had, zonder 8-11 zou de gemiddelde afwijking % verandering van de originele verwachte waarden gelijk zijn aan 1.0, de ‘perfecte score’ aangezien dit aan zou geven dat de RNN de gemiddelde verandering foutloos zou hebben voorspeld. Dit is ook de rede voor het niet gebruiken van de extra ‘aanpassings factor’ (1.4 voor de periode), dit is namelijk goed voor het gemiddelde, maar juist een stuk slechter als je dag naar dag kijkt.

Algemene informatie over de precisie van de RNN

De RNN werkt het best bij waarden die niet spontaan veranderen, spontane veranderingen zijn namelijk niet voorspelbaar, vooral als je alleen naar de voorgaande waardes kijkt. De cryptomarkt is dus een van de lastigste dingen om te voorspellen aangezien de prijs vaak spontaan en irrationeel veranderd, ook veranderd de mate waarin de waarde veranderd sterk, een dag kan het 10% veranderen en de ander 0.05%, het is dus lastig om in te schatten hoe sterk een verandering zal zijn. De RNN werkt grofweg met de gemiddelde verandering, als de verandering dus rustiger is dan gemiddeld zal de RNN de verandering in % overschatten en als de waarde opeens heel sterk veranderd zal de waarde in % onderschat worden.

Mogelijke verbeteringen:

Meer info meenemen in de berekening (volume, andere crypto waarden, nieuws, enz.)

Het model nog preciezer trainen (i.p.v. waarden per dag, waarden per minuut)

De gewichten in het model verder aanpassen (sneller reageren, langzamer reageren, waarden sterker of zwakker meetellen bij omrekening naar $ waarden)

Het model meer ‘neuronen’ geven (maakt het minder efficiënt, maar preciezer)

Ethica

Het voorspellen van de beurzen gebeurd inmiddels al een tijdje, maar vooral door grote organisaties. Een RNN die de beurs kan voorspellen is op zich niet erg nieuw, wat wel bijzonder is, is dat mijn RNN zonder enige hulp gemaakt is, dit toont aan dat je ook de beurs kunt voorspellen zonder onderdeel te zijn van een groot bedrijf. Je zou kunnen zeggen dat er nu een ‘even playingfield’ field is, oftewel dat het nu eerlijker is geworden. Het probleem is echter dat mensen nu hun gedrag aan gaan passen aan de voorspellingen, dit kan dus weer zorgen dat de voorspellingen niet meer kloppen. Verder kan het ook zorgen dat de beurs ineens kan instorten als de RNN een dip voorspelt, aangezien short-term traders dan hun geld niet willen verliezen en liever in iets stoppen dat die dag omhoog gaat. Er moet dus voorzichtig omgegaan worden met deze voorspellingen, maar het valt moeilijk tegen te houden.

Uitleg termen

Ethereum – Een digitaal platform waarop applicaties gemaakt kunnen worden en betalingen mee verricht kunnen worden.

Ether – De munt die gebruikt voor in het ethereum ecosysteem

Machine learning – Een computer iets laten leren door het veel data te laten analyseren

Neural network – Een vorm van machine learning waar de computer leert door informatie te analyseren en zogenaamde ‘weights’ aan te passen tussen verschillende lagen van het netwerk. De weights werken als gewichten die aangeven hoe sterk bepaalde kenmerken mee moeten tellen. De lagen zelf bestaat uit een digitale versie van groepen neuronen die bepaalde kenmerken leren herkennen. Het resultaat van dit netwerk is een getal dat aan kan geven hoeveel ether in de toekomst waard is.

RNN (recurrent neural network) – Een vorm van neural network waarin data in een verdere laag informatie terug kan koppelen aan voorgaande lagen

LSTM (long short-term memory) – Een manier waarmee data in een RNN opgeslagen kan worden om later hergebruikt te worden. Het lijkt op de cache van een cpu waar belangrijke data ‘in blijft hangen’, maar ook vervangen kan worden.

Logboek

A close up of text on a white background

Description generated with high confidence

A picture containing text, receipt

Description generated with very high confidence

4/3/18

Maken informatica versie verslag

2 uur

Plan van aanpak

Er staan geen datums bij omdat dit bij mij juist zorgt dat het project langzamer verloopt en ik wel weet dat ik het op tijd af krijg. Verder werd ik pas op de hoogte gesteld van deze eis nadat ik het project al zo goed als af had.

Leren hoe je een neural network moet maken

Een programma schrijven dat de data voor het nural netwerk kan ophalen en leesbaar kan maken

Maken eerste versie neural network

Verbeteren neural network

Moeilijkheden

Er waren maar twee moeilijkheden bij dit project.

Als eerste dat door de complexe werking van het netwerk het moeilijk is om je voor te stellen wat er precies gebeurt en ik hierdoor moeite had met erachter komen hoe ik de data voor de RNN begrijpbaar moest maken.

De tweede moeilijkheid was het flexibiel maken van de software die de gegevens moet ophalen voor het netwerk, tijdens het project is de structuur van de data die ik ophaalde namelijk regelmatig veranderd. Dit komt doordat de server waar ik het vandaan haal de volgorde van de data regelmatig wijzigde en ook de locatie regelmatig wijzigt.

Eindadvies

Maak een nuraal netwerk, het is leuk, leerzaam en met de juiste software en uitleg niet heel moeilijk om te maken (zolang je niks ingewikkelds wilt maken en genoeg tijd hebt).

Tips voor overname

Update het ophaal script regelmatig, sommige problemen zijn niet te voorspellen.

Handleiding

Zie demo filmpje en de keras website.