

Բովանդակություն

Նախաբան	3
Ներածություն	4
Անումալիաներ.....	5
Ռեկուրենտ նեյրոնային ցանցեր	6
Անհետացող գրադիենտ.....	9
Long Short Term Memory networks.....	10
LSTM-ի կիրառությունը Դոու-Ջոնսի արդյունաբերական ցուցանիշի ժամանակային շարքերի վրա:.....	14
Եզրակացություն	16
Գրականության ցանկ	17
Հավելված.....	18

Նախաբան

Աշխատանքի ընթացքում կիրառվել են ռեկուրսային ներդրումային ցանցեր, որոնց միջոցով կանխատեսվել են ցուցանիշներ և հայտնաբերվել են անոմալիաներ: Աշխատանքը կատարվել է Դոու-Ջոնսի արդյունաբերական ցուցանիշի ժամանակային շարքերի վրա:

Ներածություն

Խնդրի դրվածքը կայանում է նրանում, որ ունենալով ժամանակային շարք պետք է բացահայտվեն այնպիսի երևույթներ, որոնք ինչ որ չափով հանդիսանում են անոմալիա:

Անոմալիաները բացահայտելու համար կան մեքենայական ուսուցման բազմաթիվ եղանակներ:

- Unsupervised անոմալյաների բացահայտման եղանակների ժամանակ, հայտնաբերում են անոմալիաները անպիտակ տվյալների վրա այն պայմանով, որ տվյալների մեծամասնությունը նորմալ է, և փորձում է գտնել տվյալներ, որոնք չեն համապատասխանում տվյալների ընդհանուր վարքին:
- Supervised անոմալիաների բացահայտման եղանակների ժամանակ, ալգորիթմները սովորում են պիտակավորված տվյալների վրա, և նոր տվյալների առկայության դեպքում ստուգում են թե որ պիտակին է համապատասխանում:
- Semi-supervised անոմալիաների բացահայտման եղանակների ժամանակ, կառուցվում է մոդել տվյալների հիման վրա, այնուհետև կառուցված մոդելի հիման վրա ստուգում է ամեն տվյալի վարքը:

Կան մի շարք հայտնի ալգորիթմներ անոմալիաներ հայտնաբերման համար օրինակ՝

- One-class [support vector machine](#)
- [Hidden Markov models](#)
- [k-nearest neighbor](#)

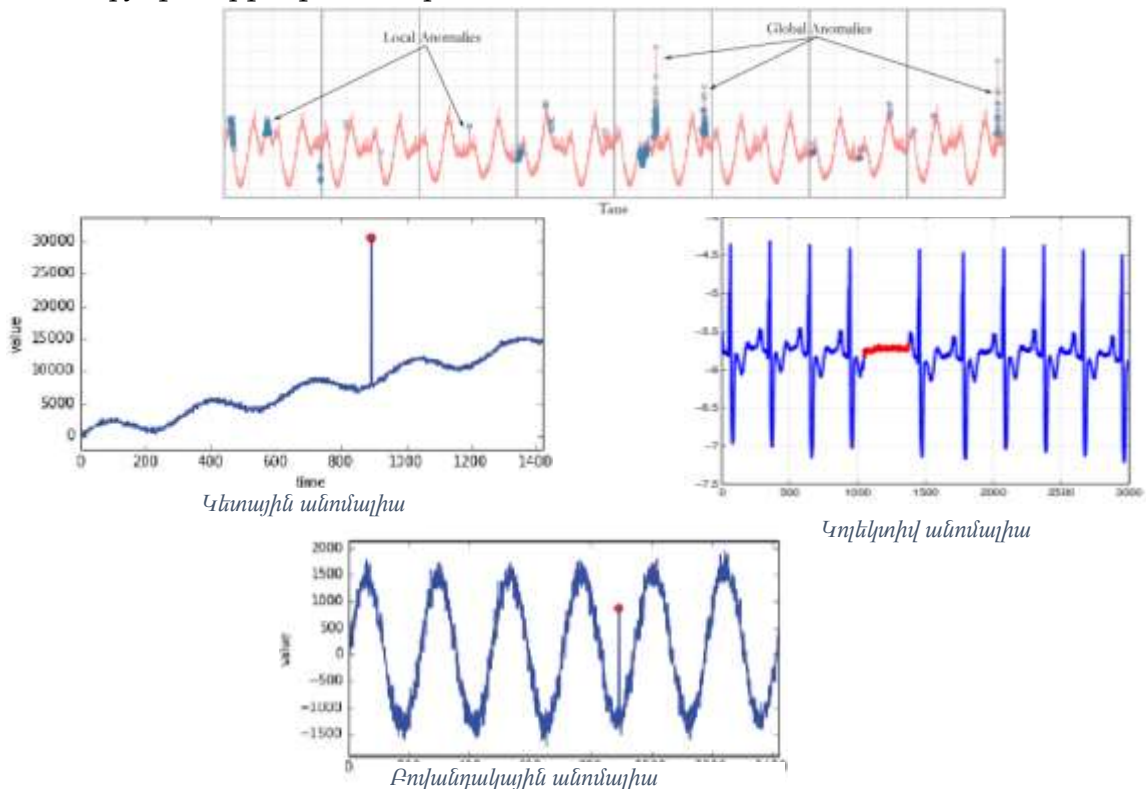
Ես իմ աշխատանքի ժամանակ կիրառել եմ Long short-term memory ալգորիթմը, որը պատկանում է արհեստական ռեկուրենտ նեյրոնային ցանցերի դասին: Այս մեթոդի շնորհիվ հնարավորություն է տրվում գտնելու բոլոր տեսակի անոմալիաները: Բոլոր օրինակները ներկայացված են ավելի պարզ խնդիրների վրա, որպեսզի ընթերցողին մատչելի լինի հասկանալու նյութը:

Անոմալիաներ

Նախ և առաջ հասկանանք ինչ է անոմալիան: Անոմալիա համարում ենք այն օբյեկտը կամ օբյեկտների խումբը, որոնք չունեն նորմալ վարքագից այլ օբյեկտների հետ համեմատած: Կամ երևույթ որը չի համապատասխանում սպասված արդյունքին:

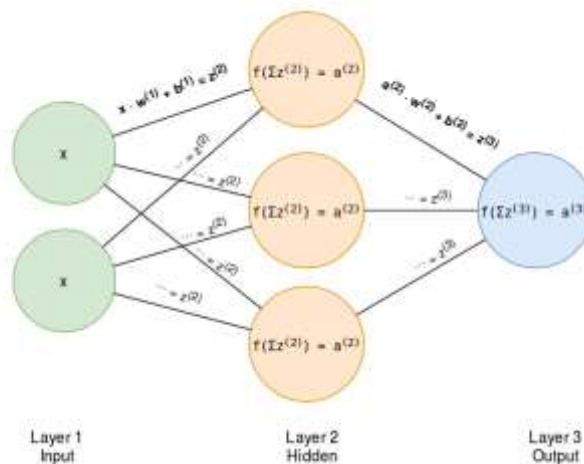
Տարբերում են անոմալիաների մի քանի տեսակներ

- Կետային անոմալիաներ: Երբ մի օբյեկտ այլ օբյեկտների հետ համեմատ կարող է համարվել անոմալիա:
- Բովանդակային անոմալիաներ: Երբ օբյեկտը չի համապատասխանում սահմանված բովանդակությանը:
- Կոլեկտիվ անոմալիաներ: Երբ օբյեկտների խումբը, այլ օբյեկտների հետ համեմատ կարող է համարվել անոմալիա:
- Լոկալ անոմալիաներ: Երբ օբյեկտը անոմալիա է իր շրջակա օբյեկտների նկատմամբ:
- Գլոբալ անոմալիա: Երբ օբյեկտը անոմալիա է իրենից հեռու գտնվող օբյեկտների նկատմամբ ևս:

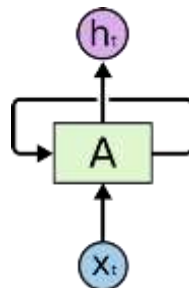


Ռեկուրենտ նեյրոնային ցանցեր

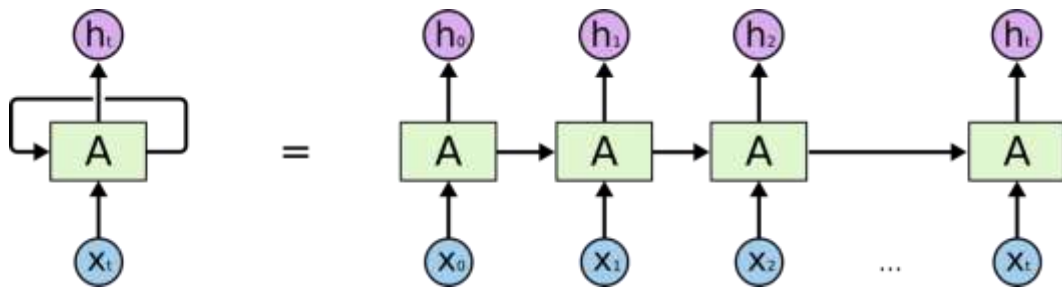
Feedforward նեյրոնային ցանցերը ունեն հետևյալ տեսքը: Նրանք ամեն անգամ որպես input վերցնում են վեկտորներ, և կատարելով որոշակի գործողություններ, վերադարձնում են որևէ արդյունք: Այսինքն ամեն անգամ մեր input տվյալները միմյանցից անկախ են:



Քանի որ մեր նպատակն է հայտնաբերել ժամանակային շարքերի անոմալիաները, հետևյալ մոդելը հարմար չէ ժամանակային շարքերը կանխատեսելու համար: Օրինակ՝ բաժնետոմսի այսօրվա գինը կախված է երեկվա գնից և այսօրվա որևէ իրադարձությունից: Այս պարագայում օգնության են գալիս Ռեկուրենտ նեյրոնային ցանցերը (այսուհետև՝ RNN): RNN-ի պարագայում նեյրոնային ցանցերը ունեն հետևյալ տեսքը:

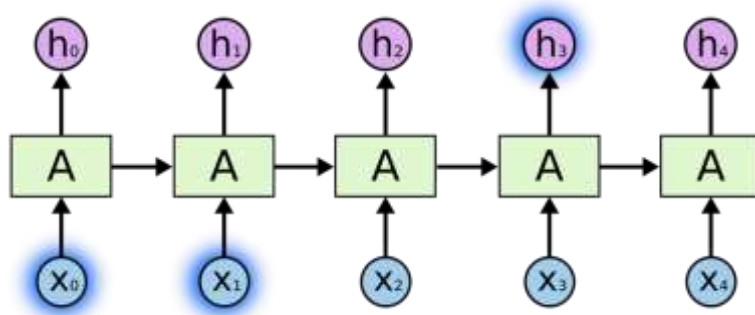


Ինչպես տեսնում ենք այն իր մեջ ներառում է ցիկլ: Ամեն անգամ նոր արդյունք վերադարձնելիս այն հաշվի է առնում նախորդ արդյունքը: Ավելի պարզ տեսնելու համար ներկայացնենք բացված տեսքով :



Ինչպես տեսնում ենք առաջին քայլում վերցնելով X_0 մեզ վերադարձնում է h_0 : Երկրորդ քայլում հաշվի առնելով նախորդ քայլում կանխատեսված h_0 -ն և նոր X_1 վեկտորը, կանխատեսում է երկրորդ ցուցանիշը:

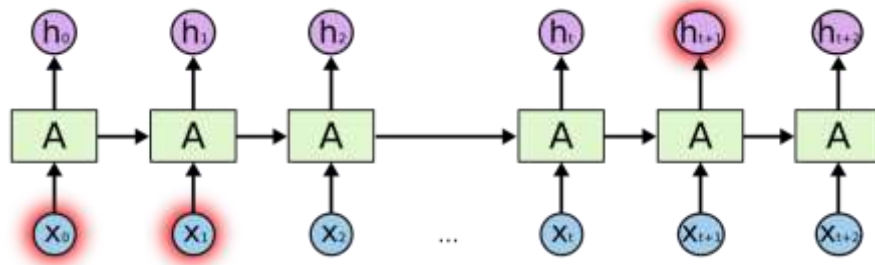
Այս ցանցը բավականին լավ է աշխատում կարճաժամկետ կանխատեսումներ կատարելու համար:



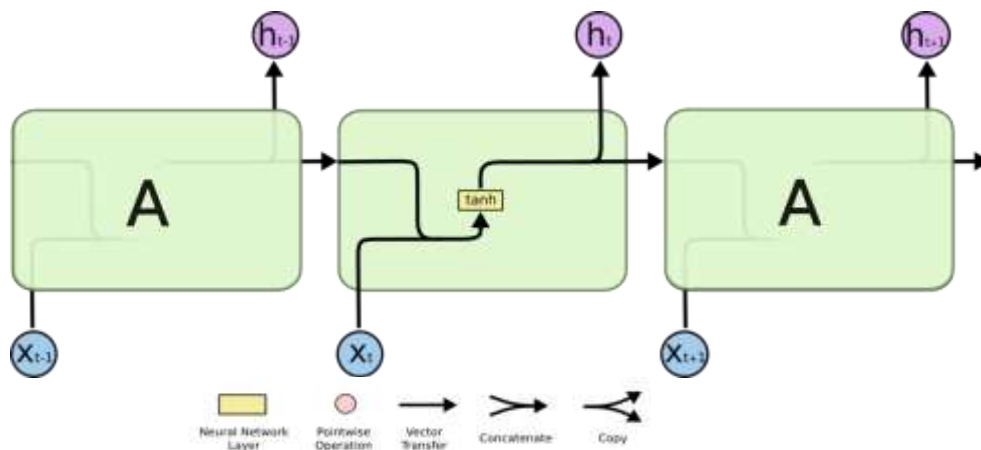
Օրինակ եթե ունենք նախադասություն «Արևը գտնվում է....» և ցանկանում ենք կանխատեսել հաջորդ բառը: Կիրառելով RNN՝ այն կվերադարձնի «երկինքում»: Այս նախադասությունը ավարտելու համար RNN-ին որևէ այլ նախադասությունից կոդմնակի ինֆորմացիա հարկավոր չէ:

Բայց լինում են դեպքեր, որ նախադասությունը ավարտելու համար պետք է նախկինից իմանալ որոշակի ինֆորմացիա:

Օրինակ՝ «Ես ծնվել եմ Հայաստանում: Իմ մայրենի լեզուն», այս պարագայում երբ շատ ենք հեռանում Հայաստան բառից, ցավոք RNN-ը ուղակի մոռանում տեքստի բովանդակությունը:



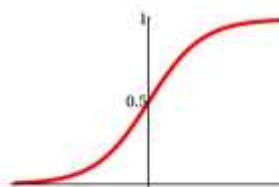
Գրեթե բոլոր ռեկուրենտ նեյրոնային ցանցերը կարող ենք ներկայացնել կրկնվող նեյրոնային ցանցերի շղթայի միջոցով: RNN-ի պարագայում այն ունի հետևյալ տեսքը:



Ինչպես տեսնում ենք շղթայի յուրաքանչյուր մոդուլ ունի միայն մեկ tanh ֆունկցիայից կազմված շերտ: Ամեն շերտում կարող են լինել տարբեր ակտիվացնող ֆունկցիաներ, բայց հաճախ օգտագործում են tanh և sigmoid ֆունկցիաները:

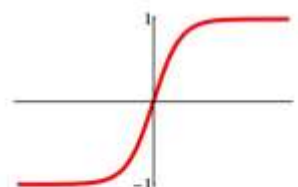
Nonlinear activation functions

$$\sigma(\Sigma) = \frac{1}{1 + e^{-\Sigma}}$$



logistic (sigmoid, unipolar)

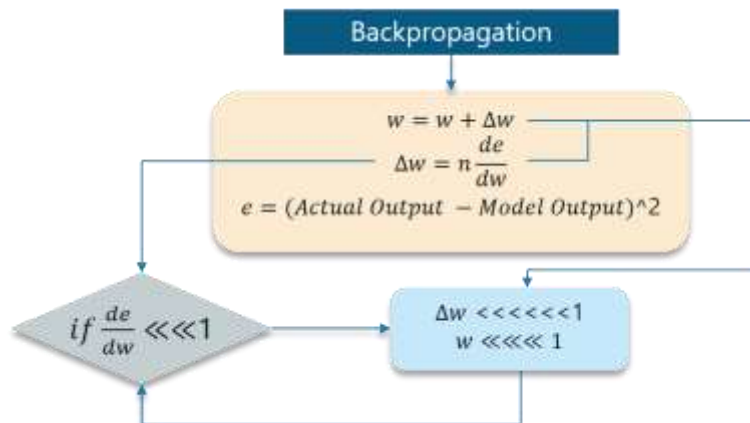
$$\tanh(\Sigma) = \frac{e^{\Sigma} - e^{-\Sigma}}{e^{\Sigma} + e^{-\Sigma}}$$



tanh (bipolar)

Անհետացող գրադիենտ

RNN ալգորիթմը սովորում է Backpropagation Through Time ալգորիթմի հիման վրա, և դրանից է առաջանում վերոնշյալ խնդիրը:

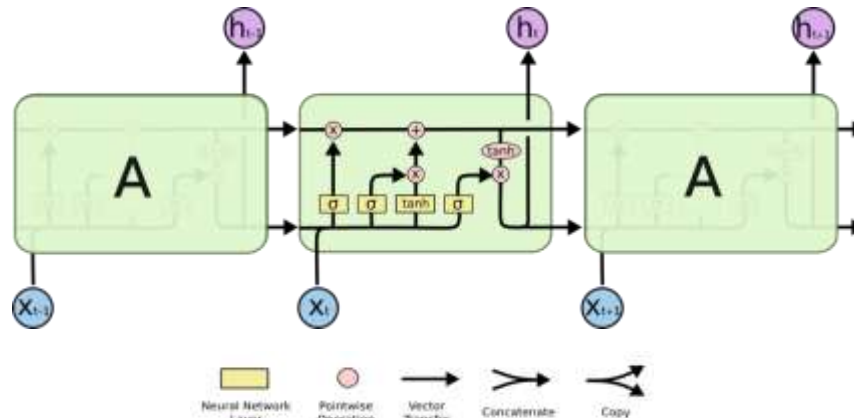


Այս ալգորիթմի ժամանակ ամեն քայլում հաշվում ենք սխալանքը (e), որը իրենից ներկայացնում է իրական ցուցանիշի և մոդելի վերադարձված ցուցանիշի տարբերության քառակուսին: Այնուհետև հաշվում ենք սխալանքի փոփոխությունը կախված կշիռների փոփոխությունից, և դա բազմապատկելով սովորելու արագությամբ (n) կստանանք թե ինչքանով պետք է փոխենք կշիռները: Այս ցուցանիշը գումարելով նախկին կշիռն կստանանք նոր կշիռներ: Մեր նպատակը հանդիսանում է նվազեցնել սխալանքը: Բայց երբ հեռանում ենք տեքստի բովանդակությունից մեզ անրաժեշտ է գնալ հետ և հասկանալ բովանդակությունը: Երբ մենք հեռանում ենք բովանդակությունից, $\frac{de}{dw}$ (գրադիենտ)-ը դառնում է բավականաչափ փոքր և մենք այն բազմապատկելով սովորելու արագությամբ՝ որը հանդիսանում է թիվ 0-ից 1 շրջակայքում, ստանում ենք հին կշիռներից շատ քիչ տարբերվող՝ նոր կշիռներ: Ե՛վ ետ վերադառնալու համար շատ քայլեր կպահանջվեն: Այս երևույթը կոչվում է «Անհետացող Գրադիենտ»: Կա նաև հակառակ խնդիրը երբ $\frac{de}{dw}$ -ն բավականաչափ մեծ է լինում և նոր կշիռները հնից տարբերվում են շատ մեծ չափով: Այս երևույթը կոչվում է «Պայթող գրադիենտ»:

Long Short Term Memory networks

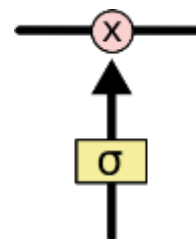
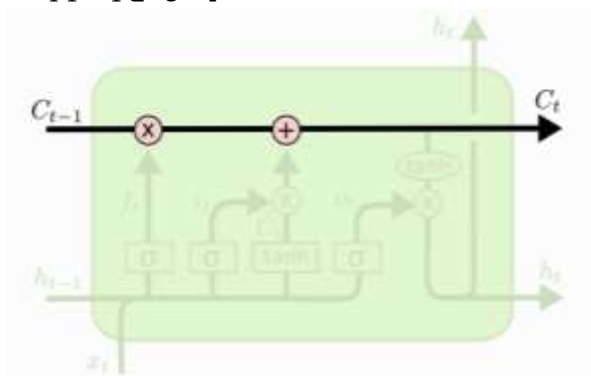
Անհետացող գրադիենտից խուսափելու մեթոդներից մեկը Long Short Term Memory networks (LSTM) ալգորիթմն է: Այս ալգորիթմը իր մեջ պահում է երկար ժամանակ առաջ եղած ինֆորմացիան:

LSTM-ի պարագայում շղթան ունի հետևյալ տեսքը:



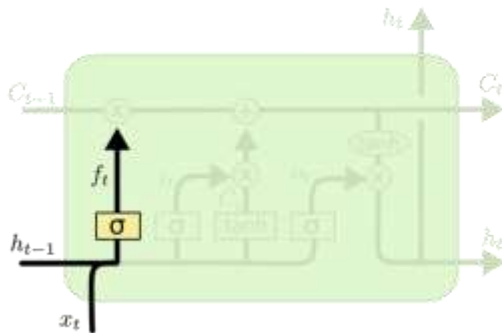
Ինչպես տեսնում ենք այն բավականաչափ հետաքրքիր է կառուցված: Այժմ դիտարկենք թե ինչպես է կառուցված մոդուլը:

Մոդուլի մեջ գտնվող հորիզոնական գիծը կոչվում է բջջի վիճակ: LSTM-ի հիմնական գաղափարը սրա մեջ է: LSTM-ի ունի հնարավորություն հեշտորեն ավելացնելու կամ պակասեցնելու ինֆորմացիա բջջի վիճակի մեջ դարպասների միջոցով:



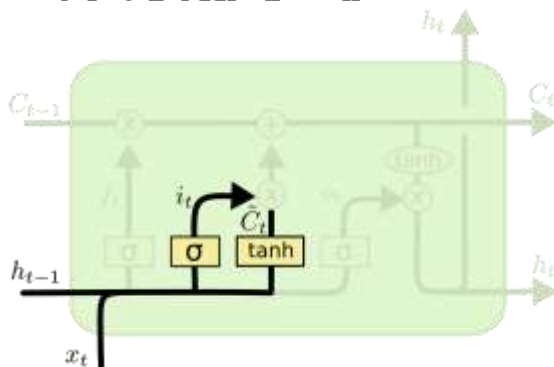
Առաջին քայլով LSTM-ը որոշում է, թե որ ինֆորմացիան պետք չէ ներառվի բջջի վիճակի մեջ: Այն կատարվում է սիգմոյիդ շերտի միջոցով որը կոչվում է “forget gate layer”: Այն միավորելով x_t -ն և h_{t-1} -ն, տալով նրանց կշիռներ և գումարելով որոշակի շեղում, տեղադրում է սիգմոյիդ ֆունկցիայի մեջ: Այն վերադարձնում է 0-ի և 1-ի միջև գտնվող վեկտոր: 1-ը նշանակում է

ամբողջապես պահիր ինֆորմացիան, իսկ 0-ն ոչինչ մի պահիր: Օրինակ՝ Արամը լավ մարդ է: Բայց Գոռը չար է: Երբ ալգորիթմը տեսնում է վերջակետը և Գոռ բառը, նա մոռանում է նախորդ նախադասության բովանդակությունը և սա կատարվում է forget gate-ի միջոցով:



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Հաջորդ քայլում որոշում է ինչ նոր ինֆորմացիա պահի բջջի վիճակի մեջ: Սա բաղկացած է երկու մասից: Առաջին քայլով սիգմոյիդ ֆունկցիայի միջոցով որոշվում է, որ արժեքները պետք է պահվեն բջջի վիճակի մեջ: Երկրորդ քայլում ստեղծել վեկտոր, որը պարունակում այն բոլոր հնարավոր արժեքներ որոնք կարող են ավելացվել բջջի վիճակի մեջ: Սա կատարվում է tanh ֆունկցիայի միջոցով, որը ընդունում է արժեքներ -1 ից 1 միջակայքում: Բազմապատկում ենք այս երկու արդյունքները և ստանում ենք կարևոր ինֆորմացիա, որը պետք է ավելացվի բջջի վիճակին:



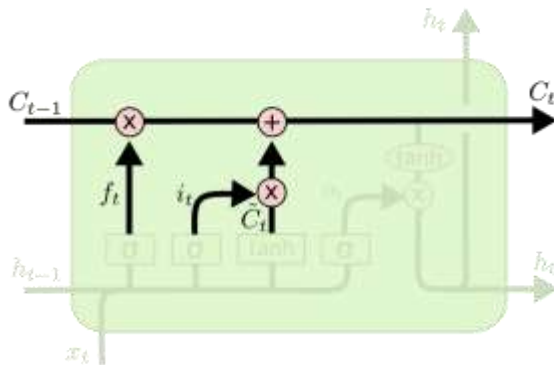
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Օրինակ՝ Արամը լողալ գիտի: Նա ինձ հեռախոսով ասաց, որ նա նավաստի է: Այստեղ կարևոր ինֆորմացիա է օրինակ ,որ Արամը լողալ գիտի և նավաստի է, բայց հեռախոսով ասելը այդքան էլ կարևոր չէ և կարող է անտեսվել:

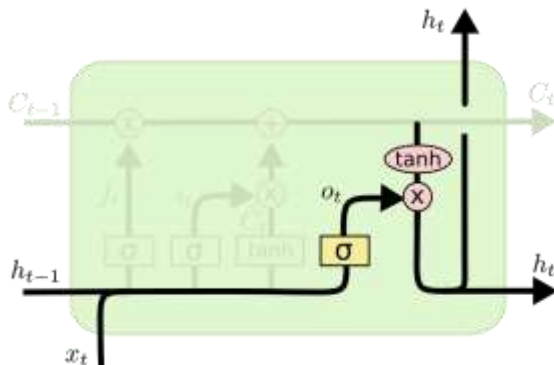
Հաջորդ քայլում պետք է C_{t-1} -ից անցնենք նոր C_t բջջի վիճակին: Բազմապատկելով նախորդ վիճակը f_t -ով ալգորիթմը մոռանում է այն, ինչ

նախորդք որոշել էինք որ պետք է մոռանալ: Այնուհետև գումարելով նախորդ արդյունքին ստանում ենք նոր բջջի վիճակ:



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Վերջին քայլում որոշում ենք output-ը: Սկզբում որոշում ենք բջջի վիճակի որ մասն ենք ներկայացնելու որպես output: Դա անում են սիգմոյիդ ֆունկցիայի միջոցով: Այնուհետև վիճակի վեկտորի վրա կիրառում ենք tanh ֆունկցիան որպեսզի ստանանք -1-ից 1 միջակայքում գտնվող թվեր: Բազմապատկելով նախորդ քայլում ստացված արդյունքի հետ, արտածում ենք այն մասը, որը որոշել էինք արտածել:



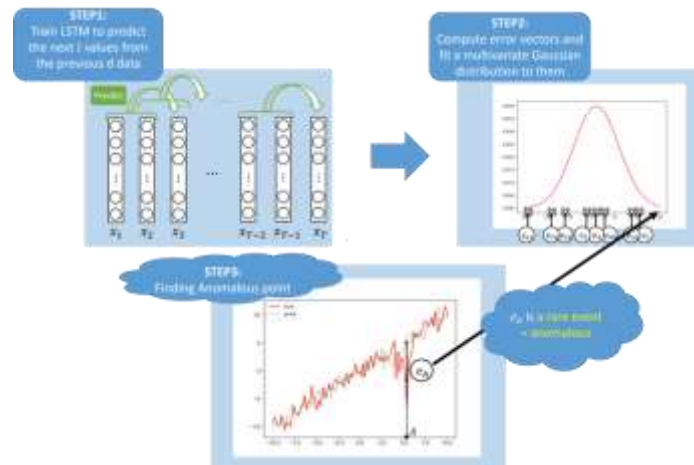
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Այժմ ցույց տանք թե ինչպես գտնել անումալյաները ժամանակային շարքերի մեջ:

Դիցուք ունենք հետևյալ ժամանակային շարքը $X = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$, որտեղ $\forall x^{(t)} \in R^m$:

Մոդելը սովորում է կանխատեսել հաջորդ 1 արժեքները նախորդ d արժեքների հիման վրա, այնպես որ $1 \leq d \leq m$: Այսպես անելով ստացվում է, որ ամեն կետի համար կատարում ենք 1 կանխատեսում: Այնուհետև ամեն կանխատեսման համար հաշվում ենք շեղումը իրական ցուցանիշից: Ստանում ենք սխալների վեկտոր $e^{(t)} = [e_{11}^{(t)}, \dots, e_{1l}^{(t)}, \dots, e_{dl}^{(t)}, \dots, e_{dl}^{(t)}]$, որտեղ $e_{ij}^{(t)}$ -ն $x_i^{(t)}$ -ի և նրա կանխատեսված արժեքի տարբերությունն է:



Այնուհետև ենթադրում ենք որ սխալները ունեն բազմաչափ Գաուսյան բաշխում $N = N(\mu, \Sigma)$: Եթե սխալը գտնվում է բաշխման պոչում, ապա կարող ենք ասել որ այն անոմալիա է: $\forall x^{(i)}$ -ի համար անոմալիաի չափը հավասար է Mahalanobis' Distance-ին: Սա չափում է, թե $x^{(i)}$ -ն ինչքան հազվադեպ երևույթ է:

$$a^{(i)} = (e^{(i)} - \mu)^T \Sigma^{-1} (e^{(i)} - \mu)$$

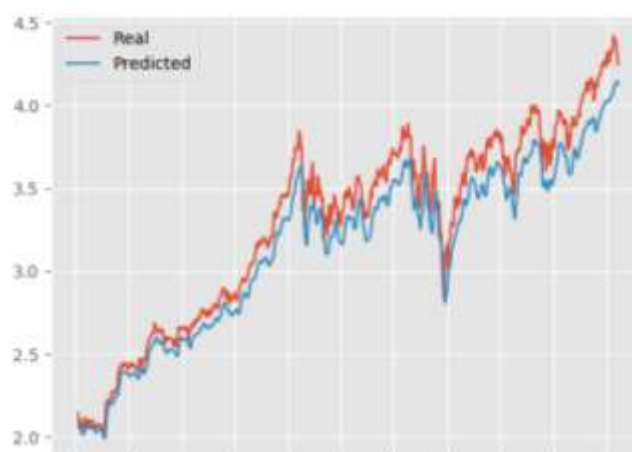
LSTM-ի կիրառությունը Դոու-Ջոնսի արդյունաբերական ցուցանիշի ժամանակային շարքերի վրա:

Ունենալով Դոու-Ջոնսի արդյունաբերական ցուցանիշի փակման գնի օրական ժամանակային շարքը՝ սկսված

1985 թվականի հունվարի 29-ից մինչև 2020 թվականի հունվարի 28-ը, փորձել եմ գտնել անոմալիաներ վերջին 10% ցուցանիշների մեջ՝ LSTM ալգորիթմի միջոցով: Այն ունի հետևյալ տեսքը:

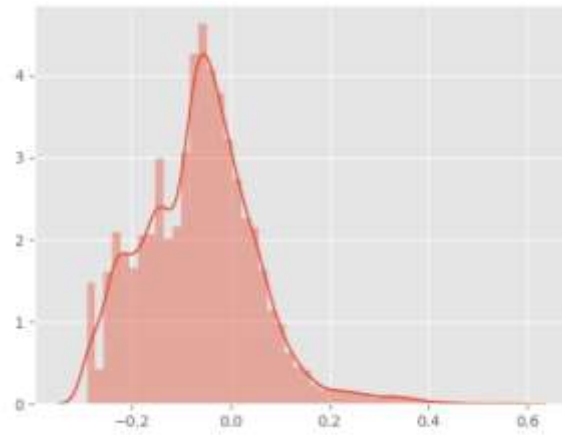


Ալգորիթմը սովորելով առաջին 7938 տվյալների վրա կանխատեսել է 882 տվյալ, ստորև ներկայացված է կանխատեսված և իրական արդյունքների գծապատկերները:

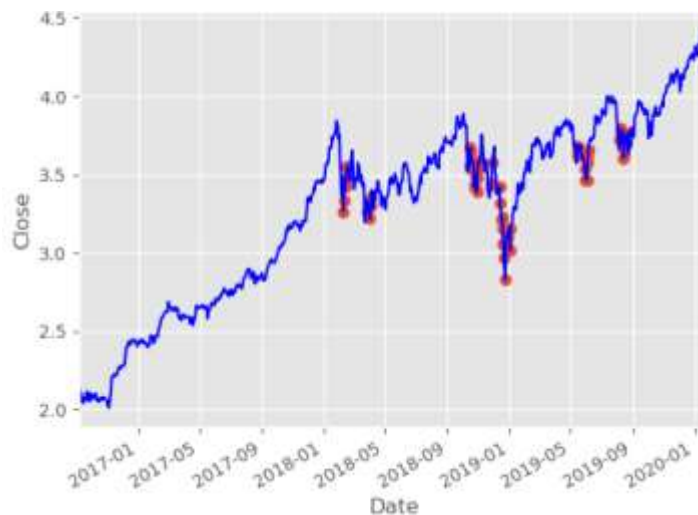


Mean squared error-ը կանխատեսված և փաստացի ցուցանիշների միջև ստացվել է 0.024024945438004087:

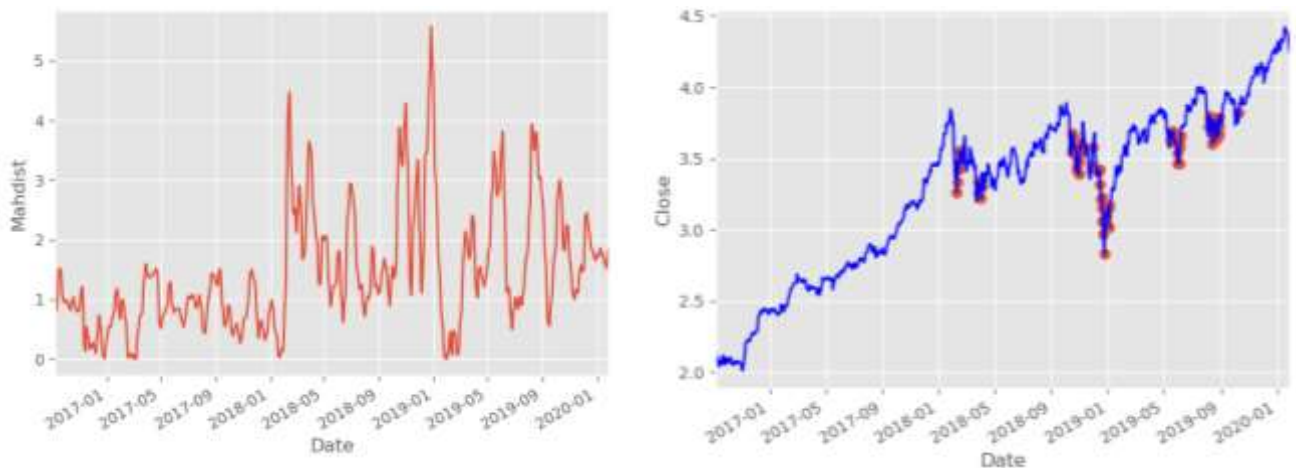
Հաշվելով իրական և կանխատեսված ցուցանիշների տարբերությունը՝
գծենք սխալանքների բաշխումը, որը ունի հետևյալ տեսքը:



Վերցնելով անոմալիայի սահմանաչափը 0.3՝ ստացվել է որ հետևյալ
կետերը հանդիսանում են անոմալիա: Ցույց տանք գրաֆիկորեն:



Այնուհետև հաշվելով Mahalanobis' Distance-ը և վերցնելով
սահմանաչափը 3, ունենք հետևյալ արդյունքը:



Եզրակացություն

Այսպիսով LSTM մոդելի միջոցով բավականին լավ կարելի է կանխատեսել բաժնետոմսերի սպասվող ցուցանիշները, ինչպես նաև գտնել այնպիսի երևույթներ, որոնք կարող են առաջացնել ցնցումներ շուկայում: Այս ալգորիթմը հնարավորություն է տալիս գտնել գլոբալ և լոկալ անոմալիաները, ինչպես նաև կոլեկտիվ և կետային անոմալիաները: Հետագայում շատացնելով ներդրոնային ցանցի շերտերը՝ կարելի է ստանալ ավելի իրատեսական կանխատեսումներ:

Գրականության ցանկ

1. Malhotra, Pankaj, et al. "LSTM-based encoder-decoder for multi-sensor anomaly detection." *arXiv preprint arXiv:1607.00148* (2016).
2. Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41.3 (2009): 1-58.
3. Shipmon, Dominique T., et al. "Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data." *arXiv preprint arXiv:1708.03665* (2017).
4. Zhu, Mengxiao, et al. "Outlier detection in sparse data with factorization machines." *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017.
5. <https://towardsdatascience.com/a-note-about-finding-anomalies-f9cedee38f0b>
6. <https://pathmind.com/wiki/lstm>
7. <http://dprogrammer.org/rnn-lstm-gru>
8. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
9. <https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/>
10. <https://www.youtube.com/watch?v=WCUNPb-5EYI>
11. <https://www.youtube.com/watch?v=y7qrilE-Zlc>

Հավելված

Աշխատանքային ֆայլ