

Recurrent Neural Networks (RNN, LSTM)

Կրիպտոարժույթների գլերը փոփոխվում են արագ և անընդհատ, ուստի հաջորդ օրվա գիշը մեծապես կախված է նախորդ օրերի տվյալներից: Ընդհանուր նեյռոնային ցանցերը (feedforward neural networks), որոնք, օրինակ, հաճախ օգտագործվում են պատկերների դասակարգման համար, չունեն հիշողություն. յուրաքանչյուր նոր տվյալ (input) մշակվում է անկախ նախորդներից: Սա լավ է աշխատում, եթե տվյալները անկախ են, բայց անհարմար է ժամանակային շարքերով աշխատելիս:

Կրիպտոարժույթների գլերի կանխատեսումը բարդ է, որովհետև հաջորդ օրվա գիշը կախված է նախորդ օրերի տվյալներից:

Recurrent Neural Networks (RNN)

RNN-ը լուծում է այս խնդիրը՝ ներառելով **hidden state** (թաքնված շերտ), որը պահպանում է նախորդ ժամանակահատվածի (time step-ի) տեղեկատվությունը

Ինչպես է գործում

Օրինակը բերեմ Bitcoin-ի գլերերով

- Նախկին օրերի տվյալները (օրինակ, վերջին 5 օրվա գլերը) ուղարկվում են ցանցին:
- RNN-ը ստանում է երկու տվյալ՝ ներկայիս օրվա գիշը և նախորդ hidden state:
- Ցանցը մշակելով ստեղծում է նոր hidden state՝ պարունակելով վերջին օրերի կարևոր ինֆորմացիան, և կանխատեսում հաջորդ օրվա գիշը:

Մաթեմատիկապես, hidden state-ը հաշվվում է այսպես՝

$$h_t = \tanh(W_x \cdot x_t + W_h \cdot h_{t-1} + b)$$

x_t – ներկայիս օրվա տվյալները

h_{t-1} – նախորդ ժամանակահատվածի (time step) փոխանցված տեղեկատվությունը

W_x, W_h – weight մատրիցներ

b – bias

RNN-ների սահմանափակումներ

Պարզ RNN-ները լավ են կարծ ժամանակային կախվածությունների համար, բայց վատ են երկարաժամկետ կախվածությունների համար, օրինակ՝ եթե ուզում ենք նախորդ ամսվա գլերի ազդեցությունը դիտարկել հաջորդ օրվա վրա, քանի որ ուսուցման ընթացքում «մոռանում ե» վաղաժամկետ տվյալները (ինչպես vanishing gradient-ի խնդիրն է կոչվում): Այս պատճառով RNN-ները չեն կարող բավարար ճշգրտությամբ սովորել երկարաժամկետ հարաբերությունները՝ անհրաժեշտ Bitcoin-ի կամ այլ կրիպտոարժույթների կանխատեսման համար:

Այս խնդիրը լուծելու համար օգտագործվում են LSTM-ներ (Long Short-Term Memory):

LSTM-ները ունեն հատուկ մեխանիզմներ՝ որոշելու, թե ինչ տեղեկատվությունները պահպանել,

ինչ մոռանալ և ինչ փոխանցել հաջորդ քայլին: Սա թույլ է տալիս ցանցին հիշել երկարաժամկետ շուկայի միտումները, ինչի շնորհիվ Bitcoin-ի կամ այլ կրիպտոարժույթների գների կանխատեսումը դառնում է ավելի ճշգրիտ:

LSTM (Long Short-Term Memory)

LSTM (Long Short-Term Memory) ցանցերը, որոնք ստեղծվել են 1997 թվականին Sepp Hochreiter-ի և Jürgen Schmidhuber-ի կողմից, լուծում են վանիշինգ գրադիենտի խնդիրը (vanishing gradient problem) իրենց հատուկ ճարտարապետությամբ:

Ցիմսական տարրերությունը պարզ RNN-ներից այն է, որ LSTM-ները օգտագործում են երկու տեսակ հիշողություն՝

Hidden state – նման է սովորական RNN-ի հիշողությանը

Cell state - թույլ է տալիս տեղեկատվությանը պահպանվել երկար ժամանակահատվածով

Gates

LSTM-ը իր հզորությունը ստանում է gates-ների շնորհիվ, որոնք թույլ են տալիս ցանցին սվորել, թե երբ հիշել, թե երբ մոռանալ և թե երբ ուշադրություն դարձնել արդեն ուսեցած հիշողություններին: Այս որոշումները մենք չենք որոշում հստակ, դրանք սովորում են ցանցը տվյալների միջոցով՝ backpropagation-ի միջոցով: (ցանց սովորելու պրոցեսը կոչվում է backpropagation)

LSTM-ը յուրաքանչյուր ժամանակային քայլում օգտագործում է **3 Gate**

1. **Forget gate** որոշում է, թե անցյալ տեղեկատվությունը ինչ չափով պահպանել կամ մոռանալ:

ընդունում է 0-1 միջակայքի արժեք՝

- 1 նշանակում է, որ համապատասխան տեղեկատվությունը պետք է պահպանվի:
- 0՝ նշանակում է, որ համապատասխան տեղեկատվությունը պետք է ամբողջությամբ մոռանալ:

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

2. **Input gate** որոշում է, թե նոր տվյալներից ինչ ավելացնել Cell state ին

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

Մենք նախ պետք է որոշենք, թե որքան է կարևոր տվյալը, իսկ հետո ստեղծենք տեղեկատվությունը Ct, որպեսզի դրանք պատրաստ լինեն հաջորդ քայլում միավորվելու:

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

Սա ստեղծում է հեև նոր տեղեկատվությունը, որը նորմալիզացվում է tanh-ով (որպեսզի արժեքները լինեն [-1, 1] միջակայքում: Այս տեղեկատվությունը դեռ չի դարձել <հիշողություն>, այլ միայն կարող է դառնալ:

Cell State-ի ԹԱՐՄԱՑՈՒՄ

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

3. **Output gate** որոշում է, թե cell state-ից որն ու ինչ չափով պետք է փոխանցվի ընթացիկ hidden state-ում:

cell state պարունակում է բոլոր երկարաժամկետ հիշողությունները, բայց ոչ բոլորն են կարևոր ընթացիկ ելքի համար: Ուստի այն առաջին հերթին անցնում է **tanh activation function**-ի միջով՝ սահմանափակելով արժեքները [-1, 1] միջակայքում: Յետո, արդյունքը output gate-ի միջոցով (որը 0–1 միջակայքում է)՝ որոշվելու, թե իրականում ինչ չափով պետք է ցույց տրվի որպես hidden state:

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o)$$

Final Hidden State

$$h_t = o_t \odot \tanh(c_t)$$

ct — cell state

ot — output gate-ի արժեքը

ht — ընթացիկ hidden state, որը փոխանցվում է հաջորդ time step-ին կամ օգտագործվում է ելքի (output) համար

LONG SHORT-TERM MEMORY NEURAL NETWORKS

