Ստոխաստիկ գրադիենտային վայրէջք

Հայկ Կարապետյան

Ունենք նկարների տվյալներ $100 \times 100 \times 3$ չափանի և նրանց քանակը 10^6 հատ: Մոդելի ուսուցման քայլերը հետևյալն են՝ ընտրել կորստի ֆունկցիա և կատարել գրադիենտային վայրեջք: Գրադիենտային վայրեջքը կատարելիս մեզ անհրաժեշտ է հաշվել բոլոր տվյալների գրադիենտները, ապա միջինացնել դրանք և թարմացնել կշիռները: Դա կհամարվի մեկ գրադիենտային վայրեջքի քայլ: 10^6 հատ տվյալների համար գրադիենտ հաշվելը բավականին ժամանակատար է, այդ պատճառով եկեք ներմուծենք նոր մեթոդ: Հաշվենք առաջին տվյալի գրադիենտը, ապա թարմացնենք կշիռները: Հաշվենք երկրորդ տվյալի գրադիենտը, ապա թարմացնենք կշիռները: Վերջում կատարած կլինենք 10^6 հատ քայլ (կշիռների թարմացում):

$$w = w - \alpha \times L_1$$
$$w = w - \alpha \times L_2$$
$$\dots$$
$$w = w - \alpha \times L_{10^6}$$
$$L_i - \text{i-րդ տվյալի կորուստը}$$
$$L = \frac{1}{n}\sum_{i=1}^{n} L_i$$

Իսկ ինչո՞ւ այս մեթոդը պետք է աշխատի: Ընդհանրապես նեյրոնային ցանցերով խնդիրներ լուծելիս մենք ենթադրում ենք, որ մեր տվյալները նույն կամ նման բաշխումներից են: Այսինքն կա ինչ որ օրինաչափություն ըստ որի նրանք ստացվում են: Այդ պատճառով, երբ մի տվյալի համար մենք մինիմիզացնում ենք կորուստը, մյուս տվյալների համար այն նույնպես փոքրանում է, քանի որ ուսուցանման ամեն քայլում մոդելը ավելի լավ ֆունկցիա է գտնում, որը տվյալների բազմությունից տանում է դեպի պիտակների բազմություն: Նկարագրված մեթոդը կոչվում է ստոխաստիկ գրադիենտային վայրէջք (stochastic gradient descent): Հետևյալ մեթոդի մեջ կարևոր է նշել, որ տվյալների ավարտվելուց հետո դրանք պատահական կերպով խառնվում են և ամեն ինչ սկսվում է նորից: Դրա նպատակը մոդելի սխալ ուսուցումից խուսափելն է: Օրինակ ունենք հետևյալ տվյալները.

$$[\text{շուն}, \text{շուն}, \dots, \text{շուն}, \text{կատու}, \dots, \text{կատու}, \text{փիղ}, \dots, \text{փիղ}]$$

Եթե մենք ստոխաստիկ գրադիենտային վայրէջքը կատարենք առանց պիտակները խառնելու, մոդելը սկզբի քայլերում ուղղակի կսովորի, որ բոլոր նկարներին պետք է ասել շուն, հետո կսովորի որ պետք է ասել կատու, հետո փիղ: Այդ պատճառով սկզբից պատահական կերպով խառնում ենք տվյալները, հետո բոլորի վրա վայրեջք կատարում, հետո նորից խառնում ենք և շարունակում: Այս մեթոդը զգայուն է այն տվյալների նկատմամբ, որոնք այնքան էլ նման չեն մեր ունեցած տվյալներին (outlier): Այդպիսի տվյալները կարող են և՛ խանգարել, և՛ օգնել: Կարող են խանգարել սխալ մինիմումի ուղղություն ցույց տալով և շեղելով իրական մինիմումի ուղղությունից: Կարող են օգնել դուրս գալ լոկալ մինիմումներից:
Outlier-ների ազդեցությունից խուսափելու նպատակով, գրադիենտային վայրէջքի քայլը կարող ենք կատարել ոչ թե մեկ տվյալի գրադիենտի հիման վրա, այլ մի քանի տվյալների գրադիենտների միջինացման միջոցով: Օրինակ՝ վերցնենք ամեն 10 տվյալի գրադիենտները միջինացնենք, հետո կատարենք կշիռների թարմացում: 10 թիվը կոչվում է batch-size, իսկ այս մեթոդը կոչվում է mini-batch գրադիենտային վայրէջք (mini-batch gradient descent):

$$w = w - \alpha\nabla\frac{1}{B}\sum_{k=(i-1)B+1}^{iB} L_k, \quad i = 1, 2, ..., \left[\frac{n}{B}\right]$$
$$B - batchsize$$

Եթե ունենք 3020 հատ տվյալ և batch-size=100, վերջին 20 հատ տվյալները կարող ենք միացնել հաջորդ քայլի 80 հատի հետ, կամ նախորդ քայլի 100 հատի հետ կամ կարող ենք առանձին միջինացնել և թարմացնել կշիռները: Mini-batch գրադիենտային վայրէջքի դեպքում նույնպես

տվյալները պատահականորեն խառնում ենք, հետո ամեն քայլի վերցնում batch-size քանակով տվյալ, միջինացնում գրադիենտները և թարմացնում կշիռները: Ստոխաստիկ գրադիենտային վայրեջքը, mini-batch գրադիենտային վայրէջքն է, երբ batch-size=1: