

# Փոխանցումային ուսուցում

## Հայկ Կարապետյան

### 1 Անհրաժեշտություն

Մեքենայական ուսուցման խնդիրներում դժվար է գտնել մեծ քանակի տվյալներ, իսկ գտնելու կամ հավաքելու դեպքում շատ ռեսուրսատար է ուսուցանել մոդելը այդքան քանակի տվյալների վրա: Առաջատար AI ընկերությունների (Google, Microsoft) համար gpu ռեսուրսը մեծ նշանակություն չունի՝ միաժամանակ միացնում ենք մի քանի ցանցերի training և տեսնում են, որն է ավելի լավ արդյունք տալիս: Իսկ թույլ համակարգիչների դեպքում ուսուցումը կարող է պահանջել շատ ցpu ռեսուրսներ, ինչպես նաև երկար ժամանակ մինչև հասնենք լավ ճշգրտության (օպտիմալ մինիմումի կետի): Այդ պատճառով կարելի է օգտագործել արդեն ուսուցանված ցանցեր և հարմարացնել մեր տվյալներին և խնդրին: Ուսուցանված մոդելի դեպքում կշիռները պատահական կերպով ընտրված չեն: Դրանք արդեն ուսուցանված կշիռներ են և ցանցի սկզբի շերտերը կարողանում են նկարից առանձնացնել low level feature-ներ (եզրագծեր, աչքի հատված, մազեր): Ուսուցումը, որի ժամանակ օգտագործում ենք արդեն ուսուցանված ցանց և այն հարմարեցնում ենք մեր տվյալներին, կոչվում է փոխանցումային ուսուցում (transfer learning): Ե՞րբ օգտագործել transfer learning:

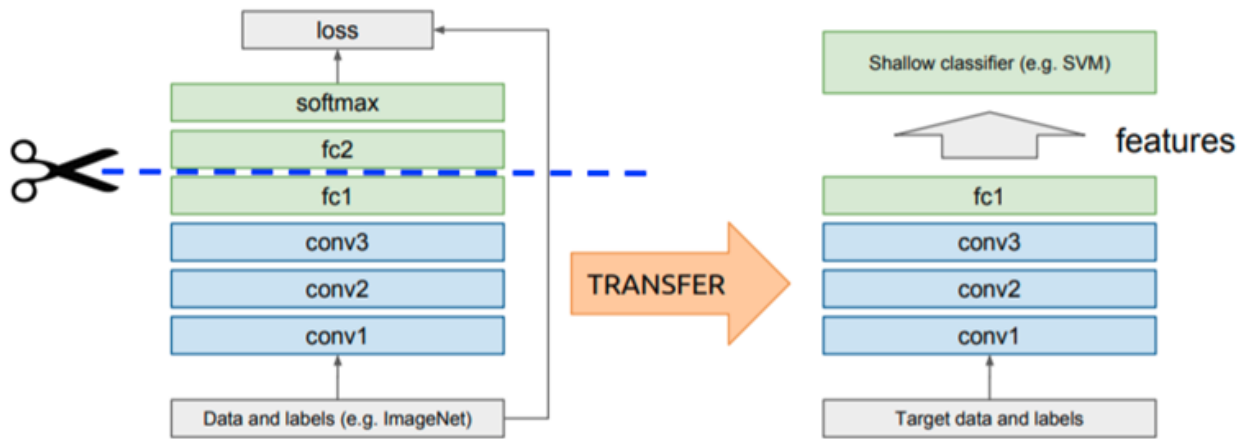
1. Երբ ուսուցանված ցանցը և մեր ցանցը ունեն նույն մոտքային տվյալները (Օրինակ՝ երկուսն էլ մոտքում ընդունում են RGB նկար):
2. Երբ ուսուցանված ցանցը ուսուցանվել է շատ քանակությամբ տվյալների վրա, իսկ մեր ցանցի ուսուցանման համար ունենք քիչ քանակությամբ տվյալներ:
3. Ուսուցանված ցանցի low-level feature-ները կարող են օգնել մեր ցանցին ուսուցանման ընթացքում:

Այստեղ գլխավոր պայմանը նման low-level feature-ներ ունենալն է: Եթե երկու ցանցն էլ մոտքում ընդունում են նկար և կատարում դասակարգում կարող ենք ենթադրել, որ նրանց low-level feature-ները նման են: Օրինակ՝ ուսուցանված ցանցը սովորել է դասակարգել 1000 class և մենք ցանկանում ենք կատարել նկարի մեջ մարդու segmentation (ամեն պիքսելը կա՛մ մարդու մի մասն է ,կա՛մ ոչ): Այս դեպքում 1000 class դասակարգող ցանցի low-level feature-ները սովորել են ընդգծել որոշ հատվածներ կամ առանձնացնել եզրագծեր և դրանք օգտակար կլինեն մարդու segmentation անելիս, քանի որ այս դեպքում նույնպես մարդու եզրագծերը մեծ նշանակություն ունեն:

### 2 Եղանակներ

#### 2.1 Fixed feature extractor

Առաջին եղանակը fixed feature extractor-ն է: Այս դեպքում ցանցի մի մասը կտրում ենք, ներքևի մասը թողնելով ֆիքսված, իսկ վերևի մասից ավելացնում ենք այն շերտերը, որոնք ցանկանում ենք և ուսուցանում ենք այդ շերտերը: Օրինակ՝ ունենք ImageNet-ի վրա ուսուցանված VGG ցանցը: Կտրում ենք դրա վերջին շերտը (1000 class softmax) և փոխարինում ենք մեր ուզած շերտերով (2 class softmax՝ մարդ կա, մարդ չկա): Վերջին շերտի պարամետրերը ուսուցման ընթացքում կհարմարվեն մեր տվյալներին, իսկ մնացած շերտերի պարամետրերը ամբողջ ուսուցման ընթացքում թողնում ենք նույնը: Գաղափարն այն է, որ VGG ցանցը սովորել է նկարից առանձնացնել պետքական feature-ներ և մենք մեր classification-ը կատարում ենք այդ feature-ների վրա: Վերջին շերտի փոխարեն կարող ենք վերջից կտրել քանի շերտ ուզում ենք և կտրած շերտերը կարող ենք փոխարինել կա՛մ նոր շերտերով, կա՛մ կարող ենք կիրառել SVM classifier (Նկար 1):



Նկար 1: Վերջին երկու շերտը կտրում ենք և փոխարինում SVM classifier-ով

Երբ տվյալները քիչ են, մեքենայական ուսուցման ալգորիթմները կարող են ավելի լավ արդյունք ցուցաբերել, քան նեյրոնային ցանցերը: Նկար 1-ի դեպքում դիտարկում ենք, որ մեր տվյալները քիչ են և կտրած շերտերը փոխարինում ենք SVM classifier-ով:

## 2.2 Fine-tuning

Երկրորդ եղանակը fine-tuning-ն է: Այս դեպքում ցանցի կառուցվածքը չենք փոփոխում և ունենք ցանցը ուսուցանելու երեք տարբերակ.

1. Ուսուցանենք ամբողջ ցանցը
2. Ուսուցանենք միայն վերին շերտերը
3. Ուսուցանենք բոլոր շերտերը տարբեր learning rate-երով

2 և 3 կետերը մոտիվացված են նրանից, որ ցանցի ներքևի շերտերը սովորում են ավելի ընդհանուր feature-ներ: Տարբեր ուսուցման արագություններով fine-tuning-ի ժամանակ սկզբի շերտերի learning rate-երը վերցնում ենք փոքր, իսկ վերին շերտերինը գնալով մեծացնում ենք: Դիտարկենք մի քանի տվյալների դեպքեր և ինչպիսի transfer learning կկիրառենք այդ դեպքում: Օրիգինալ տվյալներ ասելով հասկանում ենք այն տվյալները որոնց վրա ուսուցանվել է ցանցը մինչ transfer learning անելը (ImageNet): Նոր տվյալները մեր խնդրի համար առկա տվյալներն են:

1. Նոր տվյալները շատ են և տարբերվում են օրիգինալ տվյալներից (Օրինակ՝ աուդիոների սպեկտրոգրամներ): Այս դեպքում ցանցը ուսուցանում ենք 0-ից առանց ուսուցանված (pretrained) պարամետրեր օգտագործելու:
2. Նոր տվյալները շատ են և նման են մեր օրիգինալ տվյալներին (Օրինակ՝ մարդ կա, մարդ չկա դասակարգում): Այս դեպքում օգտագործում ենք fine-tuning և ամբողջ ցանցը շարունակում ենք ուսուցանել մեր տվյալների վրա:
3. Նոր տվյալները քիչ են և շատ են տարբերվում օրիգինալ տվյալներից: Այս դեպքում կարող ենք կիրառել մեքենայական ուսուցման ալգորիթմներ:
4. Նոր տվյալները քիչ են և նման են օրիգինալ տվյալներին: Այս դեպքում fine-tune ենք անում միայն ցանցի վերին շերտերը կամ օգտագործում ենք տարբեր learning rate էր:

Հայտնի ուսուցանված ցանցերի (vgg, resnet, inception, yolo) պարամետրերը՝ ImageNet տվյալների վրա, հասանելի են ինտերնետում:

### 3 Transfer Learning NLP<sup>1</sup> խնդիրներում

NLP-ին արհեստական բանականության մեջ տեքստերի դասակարգման, գեներացման խնդիրներով զբաղվող ճյուղն է: NLP-ի մեջ կա լեզվական մոդել (language model) հասկացություն: Այն ցանց է, որը հասկանում է լեզուն և մոտաքային տեքստի դեպքում վերադարձնում է հաջորդ բառը: Կրկնենք հետևյալ գործողություն՝ ստացված բառը ավելացնելով սկզբնական տեքստին, նորից անցկացնենք ցանցի միջով և այդպես մի քանի անգամ: Արդյունքում կստանանք սկզբնական տեքստի շարունակությունը: Ավելի մանրամասն լեզվական մոդելների մասին կխոսենք ռեկուրենտ ցանցերի և transformer-ի դասերին:

NLP խնդիրներում սովորական կերպով transfer learning կիրառելիս, այն լավ արդյունք չէր ցուցաբերում: Այսինքն երբ վերցնում ենք մեր ուսուցանված տեքստը դասակարգող մոդելը և հետո այն ուսուցանում ենք մեր տվյալների վրա (ընդ որում լեզուները կարող են տարբեր լինել), այն լավ արդյունք չէր ցուցաբերում: Դրա համար 2018 թվականին մտածել են NLP խնդիրներում transfer learning կիրառելու նոր մեթոդ: Այն բաղկացած է երեք հատվածից:

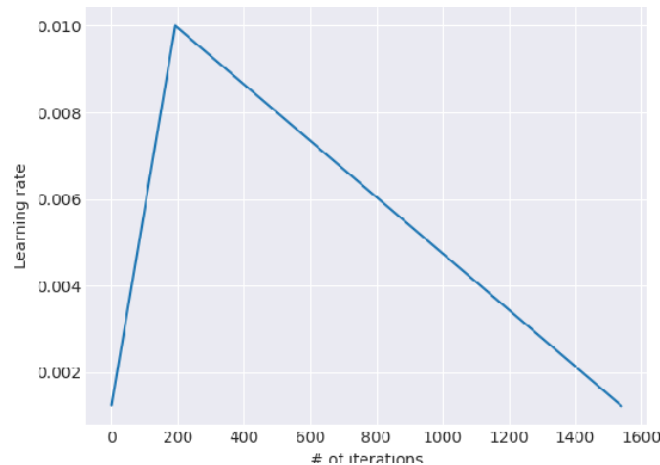
1. Ուսուցանել հաջորդ բառը գուշակող մոդել, մեր լեզվի համար (LM pre-training)
2. Fine tune անել pre-trained LM մոդելը մեր ունեցած տվյալների վրա, հաջորդ բառը գուշակելու համար
3. Fine tune անել ստացված LM մոդելը դասակարգման տվյալների վրա

Դիտարկենք մեկ օրինակ երբ կարող ենք կիրառել այս քայլերը: Պետք է ասենք նորությունը հայեմետ է թե ոչ և դրա համար ունենք հավաքած տեքստեր և դրանց համապատասխան պիտակները (1՝ հայամետ, 0՝ ոչ հայամետ): 1. Սկզբից ուսուցանում ենք կամ վերցնում ենք պատրաստի հայերեն LM: Այսինքն LM-ին մոտաքային հայերեն տեքստ տալիս, այն պետք է վերադարձնի այդ տեքստի հաջորդ բառը: Դրա ուսուցանման համար մեզ անհրաժեշտ են բավականին շատ տվյալներ և ռեսուրսներ: 2. Երկրորդ քայլով այդ LM մոդելը fine-tune ենք մեր տեքստերի վրա, որպեսզի մեր տեքստի մի մասը տալիս, այն կարողանա գուշակել հաջորդ բառը: Դա արվում է նրա համար, որովհետև մեծ քանակի տվյալների վրա ուսուցանված մոդելը կարողանում է լավ գուշակել հաջորդ բառը շատ դեպքերում, բայց եթե մեր տվյալներում առկա լինեն շատ տերմիններ, այն կարող է վատ աշխատել: 3. Երրորդ քայլով մեր fine-tune արած LM մոդելը fine-tune ենք անում արդեն մեր ունեցած դասակարգման տվյալների վրա: Այսինքն վերջին շերտում կունենանք երկու նեյրոն:

Երեք քայլից բաղկացած transfer learning-ը առաջին անգամ հայտնվել է ULMFiT գրադարանում 2018թ.-ին և ցուցաբերել է լավ արդյունք NLP խնդիրներում: Ինչպես կարող ենք ներկայացնել երեք քայլից բաղկացած transfer learning-ը նկարների խնդիրներում: Օրինակ՝ պետք է լուծենք մարդու segmentation-ի խնդիրը: 1. Վերցնում ենք ImageNet-ի վրա ուսուցանված դասկարգող ցանց: 2. Ուսուցանում ենք տարբերակելու համար նկարում մարդ կա, թե ոչ: 3. Ուսուցանում ենք segmentation ցանց: Նկարի որ պիքսելն է պատկանում մարդուն, որը ոչ (pixel wise classification): Հնարավոր է փորձենք այս տարբերակով և ավելի լավ արդյունք ստանանք:

## 4 Ուսուցման արագություն

Ցանցը transfer learning-ի մեթոդով ուսուցանելիս անհրաժեշտ է լինում փոփոխել ուսուցման արագությունը: Ուսուցանված ցանցը արդեն հասել է իր մինիմումին, և մեզ անհրաժեշտ է այդ մինիմումի կետից դուրս գալ և գնալ մեր խնդրի համար ավելի լավ մինիմում: Այդ պատճառով օգտագործում են STLR<sup>1</sup> մեթոդը (Նկար 2):



Նկար 2: Եռանկյունաձև փոփոխում ենք ուսուցման արագությունը

Սկզբից ուսուցման արագությունը մեծացնում ենք մինիմումի կետերից դուրս գալու համար, ապա փոքրացնում ենք՝ ավելի լավ մինիմումի կետի հասնելու համար: Եվ այսպես կարող ենք շարունակել մի քանի անգամ կամ ամբողջ ուսուցման ընթացքում մի անգամ: Թե ինչքան պետք է մեծանա ուսուցման արագությունը և երբվանից սկսած պետք է նվազի, դրանք հիպերպարամետրեր են:

---

1. STLR (Slanted triangular learning rates) - Եռանկյունաձև ուսուցման արագություն