# Փաթույթ

Հայկ Կարապետյան

## 1 Անհրաժեշտություն

Ունենք $(200, 200, 3)$ չափ ունեցող նկար: Հաշվենք հետևյալ նկարը մեր սովորական dense neural network-ին մուտքում տալիս, քանի կշիռ է հարկավոր: Սահմանենք մեր ցանցը կազմված մի շերտից և այդ շերտում կա 100 նեյրոն: Որպեսզի մենք մուտքում տանք նկարը, մեզ հարկավոր է այն դարձնել միաչափ և այդ դեպքում մուտքային արժեքների քանակը հավասար կլինի $200 \times 200 \times 3 = 120000$: Անհրաժեշտ կշիռների քանակը հավասար կլինի $120000 \times 100 + 100 = 12000100$: Այսինքն միայն առաջին շերտում ստացանք 12 միլիոն պարամետր, որը բավականին շատ է: Սա առաջին խնդիրն է: Երկրորդ խնդիրն այն է, որ մենք ցանկանում ենք պահպանել նկարի խորությունը, հարթացնելու (flatten) փոխարեն, քանի որ մարդն էլ նկարին նայելիս ուղղակի չի տեսնում այդ թվերը, այլ հաշվի է առնում նկարի եռաչափ (3 channel) լինելը: Հետևյալ երկու խնդիրներից բխում է փաթույթային (convolutional) ցանցերի անհրաժեշտությունը:

## 2 Մաթեմատիկական բանաձև

Convolution-ը գալիս է մաթեմատիկայից: Դիտարկենք հետևյալ բանաձևերը:

1. 1D Convolution
   (a) անընդհատ դեպք
   Ունենք երկու ֆունկցիա՝ $f, g : \mathbb{R} \to \mathbb{R}$: $f$ ֆունկցիայի վրա փաթաթել $g$ ֆունկցիան սահմանվում է. Երկու ֆունկցիաների արտադրյալի ինտեգրալը, երբ ֆունկցիաներից մեկը հայելային շրջվում է և տեղափոխվում: Բանաձևը կլինի հետևյալը.

$$(f * g)(t) =: \int_{-\infty}^{+\infty} f(x)g(t-x)dx$$
$$g(x) \to g(-x) \to g(t-x)$$

   g(x) ֆունկցիան հայելային շրջում ենք x առանցքի նկատմամբ և ստանում ենք g(-x), ապա տեղափոխում ենք x առանցքի ուղղությամբ t-ով՝ g(t-x): Եկեք տեսնենք, որ $f * g = g * f$

$$(f * g)(t) =: \int_{-\infty}^{+\infty} f(x)g(t-x)dx$$
$$t - x =: k$$
$$\int_{-\infty}^{+\infty} f(t-k)g(k)d(t-k) = \int_{+\infty}^{-\infty} -f(t-k)g(k)dk = \int_{-\infty}^{+\infty} f(t-k)g(k)dk = (g * f)(k)$$

   (b) դիսկրետ դեպք
   Ունենք իրական թվերից բաղկացած երկու հաջորդականություն՝ $\{f_n\}_{n=-\infty}^{+\infty}, \{g_n\}_{n=-\infty}^{+\infty}$: Հետևյալ հաջորդականությունների փաթաթումը n կետում գրենք հետևյալ կերպ.

$$z_n =: \sum_{k=-\infty}^{+\infty} f_k g_{n-k}$$

2. 2D Convolution
   (a) անընդհատ դեպք
   Սահմանենք $f, g : \mathbb{R}^2 \to \mathbb{R}$.

$$(f * g)(t, \tau) =: \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)g(t-x, \tau-y)dxdy$$

Այստեղ նույնպես նույն կերպ կարող ենք տեսնել որ $f * g = g * f$

(b) դիսկրետ դեպք
Վերցնենք f(x) ֆունկցիան և դա թող լինի մեր նկարը (այս ֆունկցիան վերադարձնելու է տվյալ կոորդինատի պիքսելի արժեքը) ու վերցնենք w(s, t) միջուկը (kernel), որը փաթաթելու ենք մեր նկարի վրա: $s \in [-a, a],\ t \in [-b, b],\ (x, y, s, t, a, b) \in \mathbb{Z}$

$$(f * w)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} f(x, y) w(x - s, y - t)$$

Այս դեպքում մեր $f$ նկարը ֆիքսում ենք և $w$ միջուկը շարժում ենք նկարի վրայով:

# 3 Կիրառում ցանցերում

Հիմա նայենք, թեգործնականում փաթույթը, ինչպես է կիրառվում նկարի վրա: Ունենք.

$$f = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix},\ w = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix},\ f * w = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

$k_{11}$-ը ստանալու համար w միջուկը տեղադրում ենք նկարի վրա և ամեն իրար վրա ընկած անդամ բազմապատկում ենք իրարով: Վերջում բոլորը գումարում ենք իրար:

$$k_{11} = w_{11} \times f_{11} + w_{12} \times f_{12} + w_{21} \times f_{21} + w_{22} \times f_{22}$$

$k_{12}$-ը ստանալու համար միջուկը մեկ քայլ տեղափոխում ենք աջ և կատարում փաթույթի գործողությունը:

$$k_{12} = w_{11} \times f_{12} + w_{12} \times f_{13} + w_{21} \times f_{22} + w_{22} \times f_{23}$$

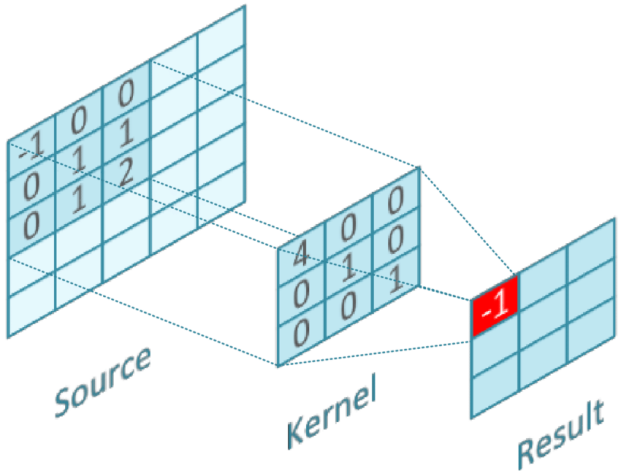
$k_{21}$-ը ստանալու համար միջուկը տեղափողում ենք տողի սկիզբը, մեկ քայլ տեղափոխում ենք ներքև և կատարում փաթույթի գործողությունը:

$$k_{21} = w_{11} \times f_{21} + w_{12} \times f_{22} + w_{21} \times f_{31} + w_{22} \times f_{32}$$

$k_{22}$-ը ստանալու համար միջուկը մեկ քայլ տեղափոխում ենք աջ և կատարում փաթույթի գործողությունը:

$$k_{22} = w_{11} \times f_{22} + w_{12} \times f_{23} + w_{21} \times f_{32} + w_{22} \times f_{33}$$

Մեկ փաթույթի քայլ պատկերված է նկար 1-ում: Հիմա նայենք, թե ինչպես ենք կիրառելու




Նկար 1: convolution քայլ

$$result_{11} = -4 + 0 + 0 + 0 + 1 + 0 + 0 + 0 + 2 = -1$$

convolution գունավոր (RGB) նկարների դեպքում: RGB նկարը gray դարձնելու համար մենք ամեն channel բազմապատկում ենք ինչ որ գործակցով և վերջում ստանում ենք մի նկար:

$$gray = \alpha_1 R + \alpha_2 G + \alpha_3 B,\ \alpha_1 + \alpha_2 + \alpha_3 = 1$$

6x6x3 նկարի վրա ուզում ենք կիրառել 3x3 չափանի միջուկով փաթույթ: 3x3 չափանի միջուկը մեզ բավարար չէ, մեզ հարկավոր է ևս մի խորություն և միջուկի չափը կդառնա 3x3x3:

$$Image = \begin{bmatrix} R \\ G \\ B \end{bmatrix},\ R = \begin{bmatrix} r_{11} & \dots & r_{16} \\ & \vdots & \\ r_{61} & \dots & r_{66} \end{bmatrix},\ G = \begin{bmatrix} g_{11} & \dots & g_{16} \\ & \vdots & \\ g_{61} & \dots & g_{66} \end{bmatrix},\ B = \begin{bmatrix} b_{11} & \dots & b_{16} \\ & \vdots & \\ b_{61} & \dots & b_{66} \end{bmatrix}$$

$$W = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix},\ W_1 = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix},\ W_2 = \begin{bmatrix} w_{11}^{'} & w_{12}^{'} & w_{13}^{'} \\ w_{21}^{'} & w_{22}^{'} & w_{23}^{'} \\ w_{31}^{'} & w_{32}^{'} & w_{33}^{'} \end{bmatrix},\ W_3 = \begin{bmatrix} w_{11}^{''} & w_{12}^{''} & w_{13}^{''} \\ w_{21}^{''} & w_{22}^{''} & w_{23}^{''} \\ w_{31}^{''} & w_{32}^{''} & w_{33}^{''} \end{bmatrix}$$

$$Image * W = R * W_1 + G * W_2 + B * W_3 = \begin{bmatrix} k_{11} & \dots k_{14} \\ & \vdots & \\ k_{41} & \dots k_{44} \end{bmatrix}$$

Տեսնենք թե ինչ է կատարվում առաջին քայլում

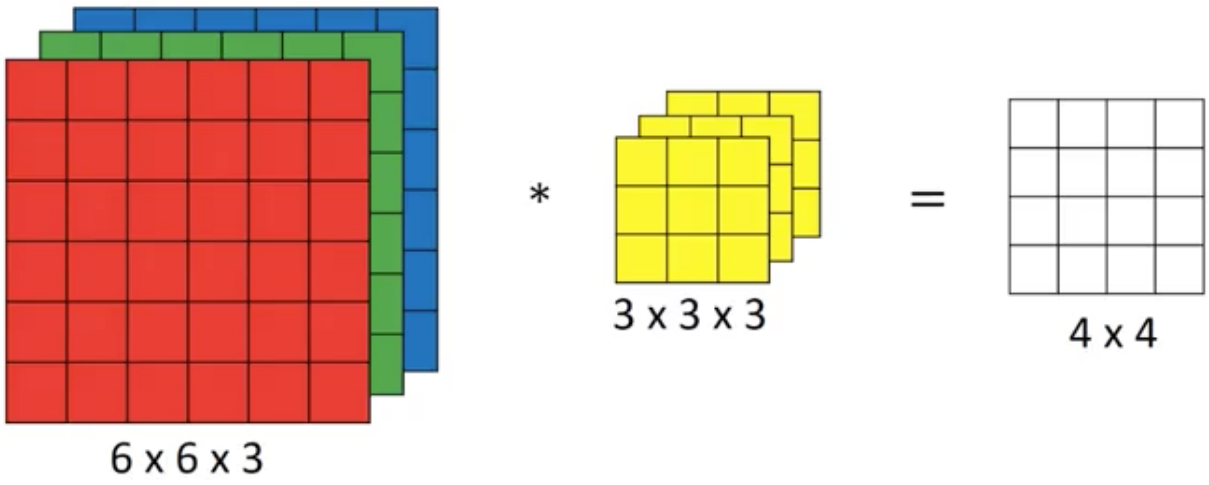
$$k_{11} = c_{11} + c_{12} + c_{13} + \dots + c_{31} + c_{32} + c_{33}$$

$$c_{ij} = w_{ij} \times r_{ij} + w_{ij}^{'} \times g_{ij} + w_{ij}^{''} \times b_{ij}\ \ i = 1,2,3,\ j = 1,2,3$$

Առաջին քայլի ժամանակ տեսանք, որ երեք միջուկները տեղադրվում են 3 channel-ների վրա, բազմապատկում ենք իրար վրա ընկած արժեքները և գումարում ենք ըստ խորության: Իսկ ինչ կստացվի եթե ոչ թե ուղղակի գումարենք իրար ըստ խորության, այլ բազմապատկենք ինչ որ գործակցով, հետո նոր գումարենք:

$$c_{ij} = \alpha_1 \times w_{ij} \times r_{ij} + \alpha_2 \times w_{ij}^{'} \times g_{ij} + \alpha_3 \times w_{ij}^{''} \times b_{ij}\ \ i = 1,2,3,\ j = 1,2,3$$

$w_{ij}$ արժեքը ուսուցանվող պարամետր է և այն կարող է սովորել $\alpha \times w_{ij}$-ի արժեքը, այդ պատճառով իմաստ չունի բազմապատկել ինչ որ գործակցով նոր գումարել:
Այսպիսով ստացանք, որ գունավոր 6x6x3 նկարի վրա 3x3x3 միջուկ կիրառելուց հետո ստանում ենք 4x4 նկար 2:




Նկար 2: Convolution RGB նկարի վրա

Իսկ ինչպե՞ս կարող ենք ստանալ 4x4x3 նկար: Դրա համար մեզ անհրաժեշտ է կիրառել 3x3x3x3 միջուկ և այդ բոլոր միջուկների կշիռները տարբեր են լինելու: Սրանից հետևում է, որ միջուկները ընդհանուր դեպքում կարող են ունենալ 4 խորություն.

1. նկարի երկարություն
2. նկարի բարձրություն
3. նկարի խորություն
4. արդյունքում ինչքան խորություն ենք ուզում ստանալ

Հիմնականում միջուկները լինում են կենտ չափերի քառակուսիներ: Կենտ չափը կապված է նրանով, որ այդ դեպքում այն ունի կենտրոն: Միջուկը նկարի վրա շարժելիս դիտարկում ենք ամեն պիքսելը, որպես կենտրոն և տեսնում ենք դրա շրջակայքում ինչ է կատարվում: Նկարի վրա 1x1 միջուկով փաթույթ կիրառելիս նկարի բոլոր պիքսելները կբազմապատկվեն մի թվով, դա նույնն է, որ մատրիցը բազմապատկենք թվով:

# 4 Padding

Հիմա դիտարկենք դեպք, երբ ցանկանում ենք նկարի չափը նույնը թողնել, բայց նկարի վրա կիրառել convolution. Սկզբնական նկարը 6x6 չափի է, դրա վրա 3x3 convolution կիրառելիս կստանանք 4x4: 6x6 նկարի բոլոր կողմերից ավելացնենք 0-ներ:

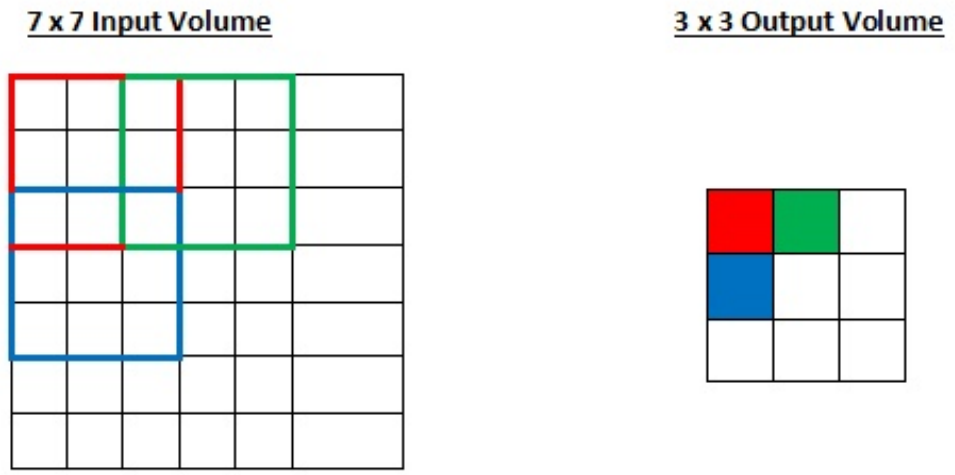
$$Image = \begin{bmatrix} x_{11} & \dots & x_{16} \\ & \vdots & \\ x_{61} & \dots & x_{66} \end{bmatrix}, \; Image\_padded = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & x_{11} & \dots & x_{16} & 0 \\ & & \vdots & & \\ 0 & x_{61} & \dots & x_{66} & 0 \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

Ստացված 8x8 նկարի վրա փաթույթ կիրառելիս կստանանք 6x6 նկար:

0-ներ նկարին ավելացնելու գործընթացը կոչվում է padding: Երբ padding=same, նշանակում է այնքան ենք ավելացնում 0-ներ, որ փաթույթ կիրառելուց հետո նկարը մնա նույն չափի, իսկ երբ padding=valid նշանակում է, որ մեզ հետաքրքիր չէ, թե նկարը ինչքան կփոքրանա մենք կիրառում ենք սովորական convolution: padding=2 նշանակում է նկարի բոլոր կողմերից ավելացնում ենք երկու սյուն կամ երկու տող 0-ներ: Padding կիրառելուց հետո նկարի եզրերը նույնպես դառնում են կենտրոններ միջուկի համար: Եթե ունենք 3x3 միջուկ և շարժում ենք սկզբնական նկարի վրայով, եզրերի արժեքները երբեք չեն գտնվի միջուկի կենտրոնում, իսկ padding-ից հետո այդպիսի խնդիր չի առաջանա և դիտարկելով եզրերի արժեքները, որպես կենտրոն կարող ենք ասել նրանց շրջակայքում ինչ է կատարվում: Այս ամենը շատ աբստրակտ է և ապացուցված չէ, որ միջուկը կենտ չափի վերցնելիս առաջանում է կենտրոն և շրջակայք հասկացություններ:

# 5 Քայլով convolution

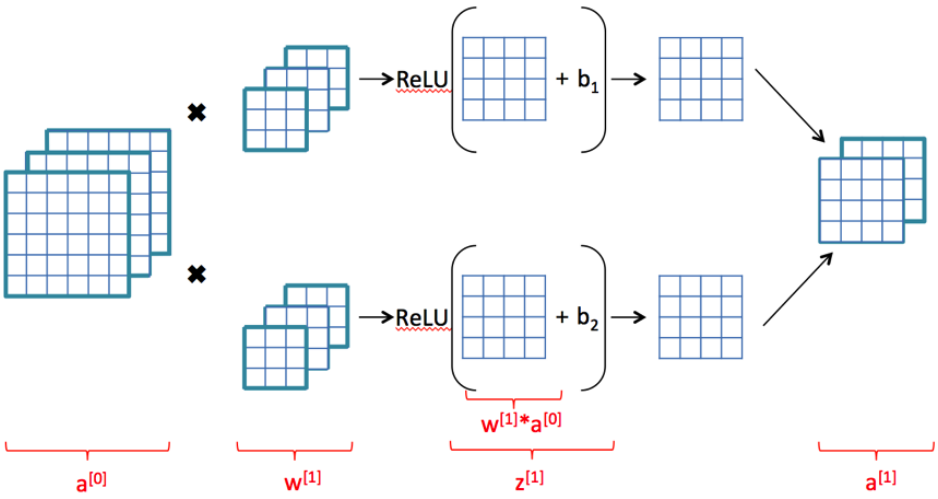
Վերը դիտարկված դեպքերում convolution-ի քայլը (stride) 1 է: Մենք դիտարկում ենք բոլոր պիքսելները (բացի եզրայիններից), որպես կենտրոն և նայում ենք դրանց շրջակայքում ինչ է կատարվում: Երբ վերցնենք և երկու քայլով շարժվենք, կդիտարկենք միայն կենտ տեղերում գտնվող պիքսելները (չհաշված եզրերը), որպես կենտրոն և կնայենք դրանց շրջակայքը (Նկար 3): Արդյունքում կստանանք ավելի փոքր նկար, քան մեկ քայլով փաթույթի դեպքում էր:




Նկար 3: 2 strided convolution

# 6 Մեկ շերտ convolution

Նկար 4-ում պատկերված է, թե ինչ տեսք ունի փաթույթի մեկ շերտը: Մեզ հարկավոր է ունենալ 3 խորություն ունեցող միջուկ, անցկացնել այդ միջուկը մեր նկարի վրայով ստանալ մեկ խորություն ունեցող մատրից, գումարել բիաս և կիրառել ակտիվացիոն ֆունկցիա: Արդյունքում ստացված մատրիցը անվանում են feature map: Այն կարող ենք տպել և կտեսնենք, որ սկզբնական նկարն է, որոշ ձևափոխություններով: Օրինակ՝ եզրերը ընդգծված կամ շան պոչի մասը առանձնացված: Մեր 3 խորությամբ միջուկը նման է մեկ նեյրոնի և եթե ուզում ենք շերտում ունենալ մի քանի նեյրոն և համապատասխանաբար արդյունքում մի քանի feature map, մեզ պետք է մի քանի 3 խորությամբ միջուկներ կիրառել և ամեն մեկին առանձին գումարել բիաս, կիրառել ակտիվացիոն ֆունկցիա և ստացված feature map-երը միավորել իրար ըստ խորության:




Նկար 4: 1 շերտից բաղկացած convolution: Շերտում առկա է երկու նեյրոն

Մեկ նեյրոնից բաղկացած convolution կիրառելու դեպքում output feature map-ի չափը որոշվում է հետևյալ բանաձևով.

$$W_{new} = \frac{W - F + 2P}{S} + 1$$

$W_{new}$ – Նկարի նոր երկարություն

$W$ – Նկարի երկարություն

$F$ – Միջուկի (kernel կամ filter) երկարությունը (3x3‘ 3)

$P$ – Padding-ի չափը

$S$ – Քայլի չափը (stride)

Նույն կերպ կարող են որոշել ստացված feature map-ի բարձրությունը։

# 7 Կիրառություն նկարներում

Ունենք նկար և մեզ պետք է նկարից առանձնացնել եզրագծերը։ Եկեք ամեն սյունից հանենք իր նախորդ սյունը։ Եթե սյունակների թվերը նույնը լինեն իրար կզրոյացնեն, հակառակ դեպքում կունենանք 0-ից մեծ թիվ և դա կնշանակի, որ եզրագիծ է։ Օրինակ երկնից դեպի գետին սահմանը կունենա եզրագիծ։ Ինչպե՞ս սա կարող ենք ներկայացնել convolution-ի տեսքով։ Վերցնենք հետևյալ միջուկը.

$$w = \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix}$$

Հետևյալ միջուկով նկարի վրա անցնելիս ամեն սյան արժեքից կհանենք իր նախորդ սյան արժեքը։ Ավելի լավ միջուկներին կարող եք ծանոթանալ հետևյալ կայքում։

# 8 Եզրակացություն

Սկզբնական խնդիրն այն էր, որ մենք սովորական dense շերտ օգտագործելու դեպքում ունենում էինք շատ ուսուցանվող պարամետրեր (12 միլիոն)։ Եկեք հաշվենք, թե ինչքան պարամետր ենք օգտագործում փաթույթային ցանցերում։ Ուսուցանվող պարամետրերը միջուկների արժեքներն են։ Այսինքն եթե ունենք 3x3 չափի միջուկ, ուսուցանվող պարամետրերի քանակը 9 է։ Վերցնենք առաջին շերտում ունենք 3x3x3 չափանի 10 հատ միջուկ։ Ընդհանուր ուսուցանվող պարամետրերի քանակը ստացվեց 270։ Փաթույթային ցանցերը նման են մարդուն։ Նկարի սկզբից միջուկը վազում է նկարի վրայով և փնտրում է իրեն անհրաժեշտ օբյեկտները։ Օրինակ՝ նկարի վրա վազելիս ամեն հատվածի կարող է ասել այդ հատվածում շան ինչ որ մաս կա թե ոչ։ Եթե ցանցը լրիվ փաթույթային է (fully convolutional), այսինքն ոչ մի dense շերտ առկա չէ, ցանցի մուտքային նկարը կարող է լինել տարբեր չափերի։ Ի տարբերություն dense շերտի հնարավոր է ունենալ ոչ ֆիքսված չափի մուտք։