# Լայնացված և տրանսպոնացված փաթույթներ։ Փաթույթը որպես մատրիցային գործողություն

Հայկ Կարապետյան

## 1 Լայնացված փաթույթ

Վերհիշենք convolution-ի մաթեմատիկական բանաձևը.

$$(F * w)(p) = \sum_{s=-a}^{a} F(s)w(p-s)$$

Նշանակենք s+t=p և կստանանք

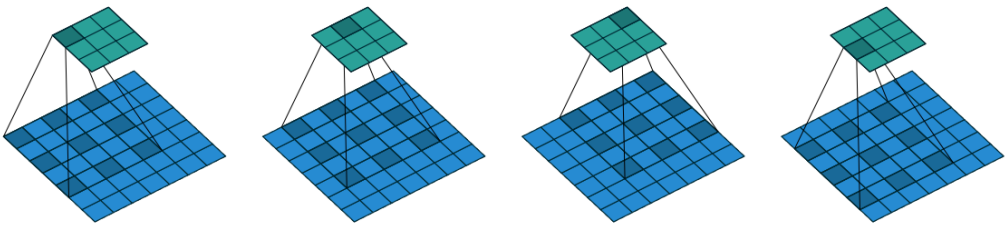
$$(F * w)(p) = \sum_{s+t=p} F(s)w(t)$$

Լայնացված փաթույթի (dilated convolution) բանաձևը հետևյալն է.

$$(F * w)(p) = \sum_{s+It=p} F(s)w(t)$$

Սա $I$-dilated convolution-ի բանաձևն է։ Երբ $I = 1$ ստանում ենք սովորական convolution-ի բանաձևը

Հիմա հասկանանք, թե հետևյալ բանաձևը ինչ է նշանակում պատկերավոր։

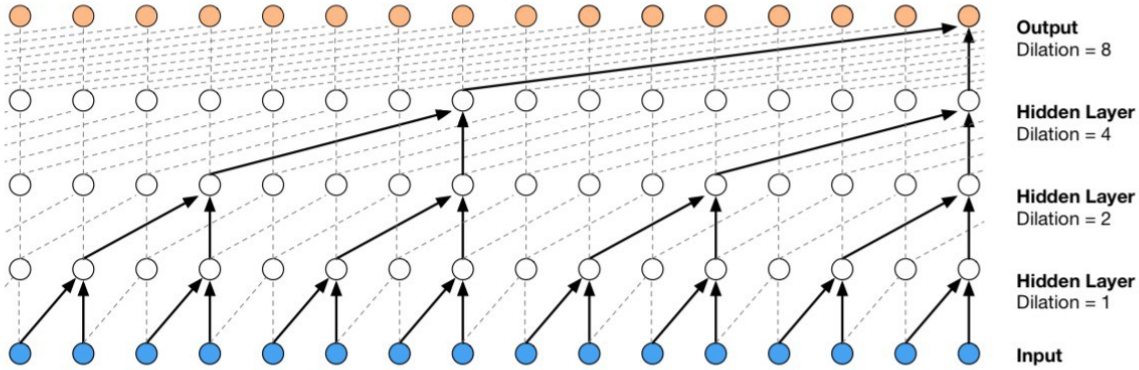


Նկար 1: 2D 2-dilated convolution-ի 4 քայլ

Նկար 1-ում պատկերված է 2-dilated convolution։ Կարող ենք ասել, որ կիրառում ենք 5x5 չափանի kernel, որտեղ մեկ ու մեջ գրված են 0-ներ։

$$w = \begin{bmatrix} w_{11} & 0 & w_{12} & 0 & w_{13} \\ 0 & 0 & 0 & 0 & 0 \\ w_{21} & 0 & w_{22} & 0 & w_{23} \\ 0 & 0 & 0 & 0 & 0 \\ w_{31} & 0 & w_{32} & 0 & w_{33} \end{bmatrix}$$

1-dilated convolution-ի դեպքում մենք մի պիքսելը ավելի քիչ անգամ ենք հաշվի առնում, քան սովորականի դեպքում, բայց վերջնական output-ը ստանալիս օգտագործում ենք բոլոր պիքսելները։ Հիմա նայենք ինչպես է փոփոխվում receptive field-ը dilated convolution օգտագործելիս։ Վերցնենք 10×10 չափի նկար։ Առաջին դեպքում երկու անգամ կիրառենք 3×3 kernel-ով convolution, իսկ երկրորդ դեպքում սկզբից կիրառենք 3×3 չափանի 2-dilated convolution և հետո կիրառենք սովորական 3×3 kernel-ով convolution։ Առաջին դեպքում ինչպես հիշում ենք receptive field-ը հավասար կլինի 5×5 կամ որ նույնն է 25։ Երկրորդ դեպքում վերջին շերտի receptive field-ը dilated convolution-ի նկատմամբ կլինի 3×3, իսկ սկզբնական նկարի նկատմամբ հավասար կլինի 7×7։ Վերջին convolution-ի ժամանակ օգտագործվել է նախորդ feature map-ի 3×3 տեղամասի պիքսելները, իսկ այդ բոլոր պիքսելները ստանալու համար dilated convolution-ը հարկավոր է երկու անգամ տեղափոխել աջ և արդյունքում կունենանք receptive field=7×7։ Նույն քանակությամբ կշիռներ օգտագործելով ստացանք ավելի մեծ receptive field։

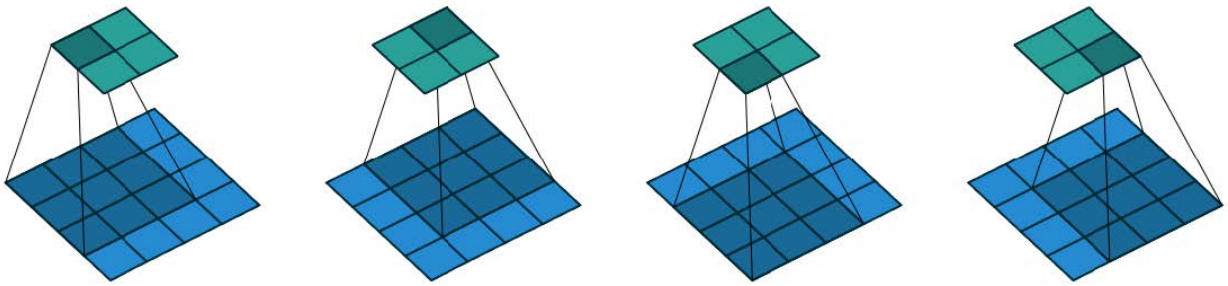



Նկար 2: 1D dilated convolution-ի օրինակ: WaveNet օգտագործելով բոլոր մուտքային արժեքները: Հաստ գծերը ցույց են տալիս receptive field-ը

Այժմ դիտարկենք 1D dilated convolution-ի դեպքը: Նկար 2-ում պատկերված է 1D dilated convolution-ի օրինակ: Շերտերում միջուկի չափը $2\times1$ է: Սկզբից կատարում ենք 1-dilated convolution, նույնն է ինչ սովորական convolution: Երկրորդ շերտում մեծացնում ենք dilation-ի չափը և դարձնում ենք 2: Այսինքն ամեն արժեքների միջև բաց է թողված 1 վանդակ: Երրորդ շերտում dilation-ի արժեքը 4 է և վերջին շերտում դարձնում ենք 8: Նկարում հաստ գծերով ցույց է տալիս, որ վերջնական output-ի մի արժեքը օգտագործում է մուտքային բոլոր արժեքները, այսինքն receptive field-ը հավասար է մուտքային արժեքների քանակին և դրանից մեծ ստանալ հնարավոր չէ: Հաշվենք կշիռների քանակը: Ամեն շերտում կա մի միջուկ $2\times1$ չափանի, այսինքն երկու ուսուցանվող պարամետր: 4 շերտի արդյունքում կունենանք 8 ուսուցանվող պարամետր:

## 2 Փաթույթը որպես մատրիցային գործողություն

Ունենք նկար 3-ում պատկերված փաթույթային գործողությունը և այն ուզում ենք ներկայացնել մատրիցային արտադրյալի տեսքով: Մեր սկզբնական նկարը $4\times4$ նկար է: Հարթեցնենք այն



Նկար 3: $4\times4$ նկարի վրա $3\times3$ convolution-ի օրինակ: Արդյունքում ստանում ենք $2\times2$ feature map

և կստանանք (16, 1) չափանի մատրից: Հարթեցնելն իրականացնում ենք հետևյալ կերպ: Վերցնում ենք առաջին տողը, կողքից ավելացնում երկրորդ տողը և այդպես շարունակ մինչև 4-րդ տողը ու ստացված (1, 16) մատրիցը տրանսպոնացնում ենք: convolution-ից հետո մեզ հարկավոր է ստանալ $2\times2$ մատրից կամ որ նույնն է (4, 1) մատրից և ձևափոխելուց (reshape) հետո կստանանք $2\times2$ մատրից: (16, 1)-ից (4, 1) ստանալու համար մեզ հարկավոր է (16, 1) մատրիցը ձախից բազմապատկել (4, 16) մատրիցով:

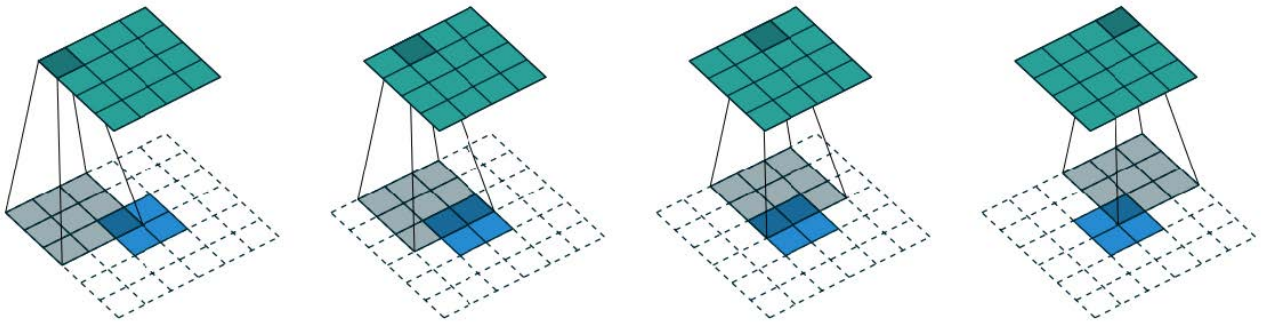
Դիտարկենք հետևյալ մատրիցը.

$$W = \begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix}$$

Նկարը ձախից հետևյալ մատրիցով բազմապատկելը նույնն է, որ դրա վրա կիրառենք 3×3 միջուկով convolution: Առաջին տողը բազմապատկում ենք (16, 1) մատրիցով և գումարում իրար: Արդյունքում նկարի առաջին տողի երեք պիքսելը կբազմապատկվեն միջուկի առաձին 3 կշիռներով, իսկ չորրոդ պիքսելը 0-ով, ապա երկրորդ տողի երեք պիքսելը կբազմապատկվի կշիռներով, 4-րդ պիքսելը 0-ով, ապա 3-րդ տողի երեք պիքսելները կշիռներով, չորրորդը՝ 0-ով: 4-րդ տողի բոլոր պիքսելները կբազմապատկենք 0-ով: Բազմապատկումներից ստացված արդյունքները կգումարենք իրար: Ստացված արդյունքը կլինի նույնը, ինչ միջուկը տեղադրեինք նկարի ձախ վերևի անկյունում: (4, 16) մատրիցի երկրորդ տողը նկարի հետ բազմապատկելիս՝ առաջին տողի, երկրորդ տողի և երրորդ տողի առաջին պիքսելները կբազմապատկվեն զրոյով, իսկ այդ տողերի մնացած պիքսելները համապատասխան կշիռներով: Բազմապատկումների արդյունքները գումարենք: 4-րդ տողը նորից կբազմապատկենք 0-ով: Ստացանք նույն արդյունքը, ինչ միջուկը մի քայլ տեղաշարժեինք աջ: Երրորդ տողը նույնն է, ինչ միջուկը մի քայլ իջեցնենք և չորրորդ տողը նույնն է, ինչ միջուկը դնենք նկարի աջ ներքևի անկյունում:

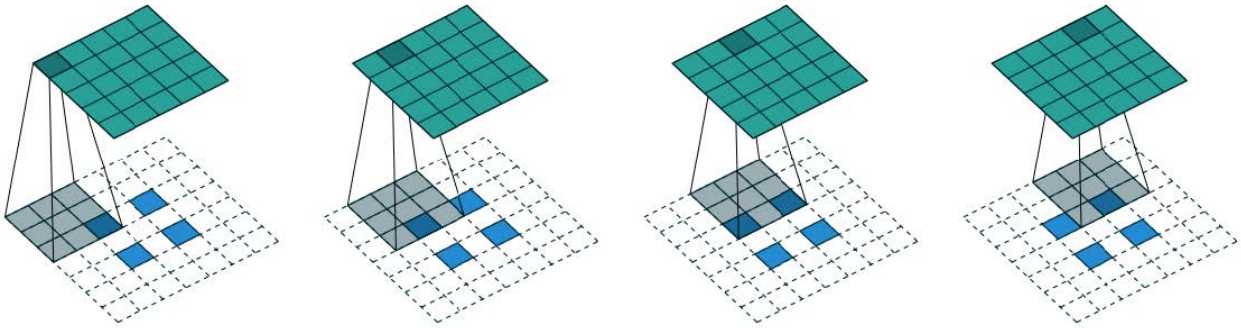
# 3 Տրանսպոնացված փաթույթ

Ունենք 2×2 չափանի նկար և դրանից ցանկանում ենք ստանալ 4×4 չափանի մատրից: Նկարը կարող ենք հարթեցնել (flatten) և ստանալ (4, 1) չափանի մատրից, իսկ վերջնական 4×4 մատրիցը նույնն է, ինչ (16, 1) մատրիցը: (4, 1)-ից (16, 1) մատրից ստանալու համար անհրաժեշտ է սկբնական նկարը ձախից բազմապատկել (16, 4) մատրիցով: Այդ մատրիցը կլինի վերը նկարագրված մատրիցը տրանսպոնացվածը: Այսինքն նկարը մեծացնելու համար, մեզ անհրաժեշտ է ունենալ այն մատրիցը, որով մեծ նկարը փոքրացնելու էինք և այդ մատրիցը կբազմապատկենք փոքր նկարի հետ: Պատկերավոր նկարի մեծացումը տեղի է ունենալու նկար 4-ում պատկերված եղանակով:



Նկար 4: Transposed convolution՝ kernel_size=3×3, stride=1:
2×2 նկարից ստանում ենք 4×4 նկար

Եթե ուզում ենք նկարը ավելի մեծացնել kenrel_size թողնելով նույնը, կարող ենք փոխել քայլը որով շարժվում ենք։ Նկար 5-ում պատկերված է տրանսպոնացված convolution-ի օրինակ, երբ քայլը=1։



Նկար 5: Transposed convolution՝ kernel_size=3×3, stride=2:
2×2 նկարից ստանում ենք 5×5 նկար

Կարող ենք տեսնել, որ ստացվում է 2-dilated convolution-ի դեպքին նման դեպք։ Դատարկ վանդակներում գրված են 0-ներ։