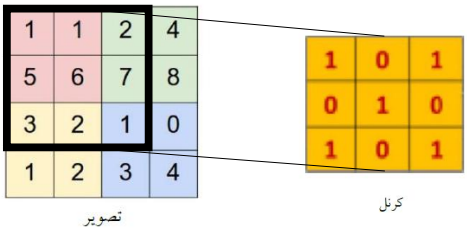
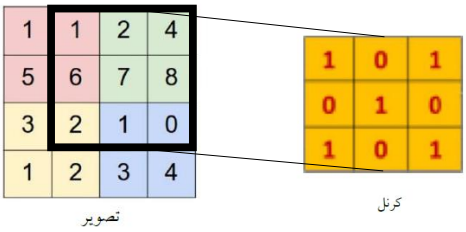
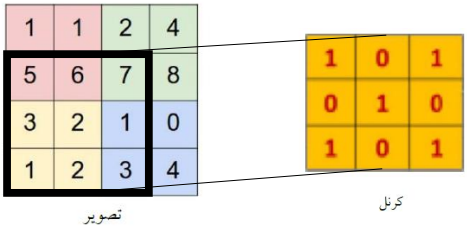
الف) نتیجه اعمال کرنل زیر بر تصویر (کانولوشن کرنل و تصویر) چیست؟ کرنل هر بار یک خانه در راستای افقی منتقل می شود.



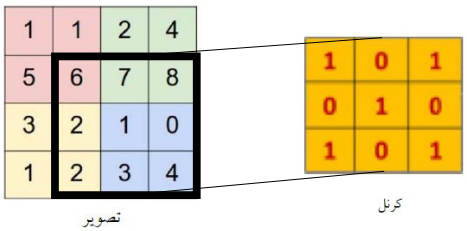
$$1(1) + 1(0) + 2(1) + 5(0) + 6(1) + 7(0) + 3(1) + 2(0) + 1(1) = 13$$



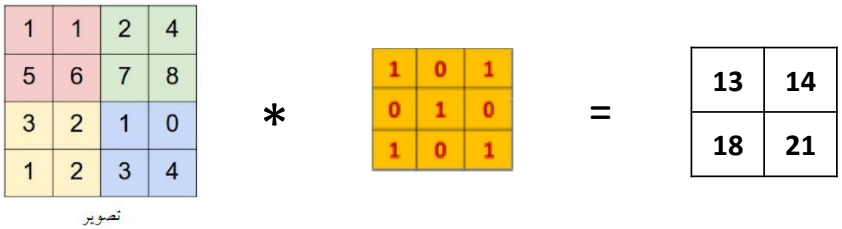
$$1(1) + 2(0) + 4(1) + 6(0) + 7(1) + 8(0) + 2(1) + 1(0) + 0(1) = 14$$



$$5(1) + 6(0) + 7(1) + 3(0) + 2(1) + 1(0) + 1(1) + 2(0) + 3(1) = 18$$



$$6(1) + 7(0) + 8(1) + 2(0) + 1(1) + 0(0) + 2(1) + 3(0) + 4(1) = 21$$

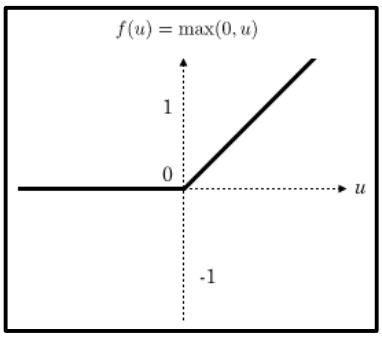


ب) نتیجه بکار بردن تابع ReLU بر خروجی قسمت الف چیست؟

دلیل استفاده از توابع فعالساز با رفتار غیرخطی (مانند ReLU) بعد از هر لایه افزایش رفتار غیر خطی شبکه می باشد. اگر از توابع فعالساز غیر خطی استفاده نشود، ساختار موجود تنها رفتار خطی خواهد شد و تنها قادر به مدل سازی توابع خطی می باشد. پس در نتیجه استفاده از شبکه عصبی با چند لایه نیز منطقی نیست چرا که می توان آن را تنها با یک لایه نیز تشکیل داد. به طور مثال، اگر X ورودی، و دو تابع خطی f_1 (لایه اول) و f_2 (لایه دوم) داشته باشیم و خروجی تابع اول به عنوان ورودی تابع دوم در نظر گرفته شود (مانند ساختارهای عصبی)، مطابق رابطه زیر مشاهده می شود که میتوان هر دوی این توابع را ادغام و یک تابع کلی خطی ساخت که نشان می دهد تنها یک لایه کافی است:

$$f_1(X) = aX$$
$$f_2\big(f_1(X)\big) = bf_1(X)$$
$$f_2\big(f_1(X)\big) = (ab)X$$

$$ReLU\left(\begin{bmatrix}13 & 14\\ 18 & 21\end{bmatrix}\right) = \begin{bmatrix}13 & 14\\ 18 & 21\end{bmatrix}$$



ReLU Activation Function

ج) در صورتیکه بر روی نتیجه قسمت (الف) و (ب) عملیات max pooling با سایز ۲ و گام ۲ صورت گیرد، نتیجه چه خواهد بود؟

13	**14**
18	**21**

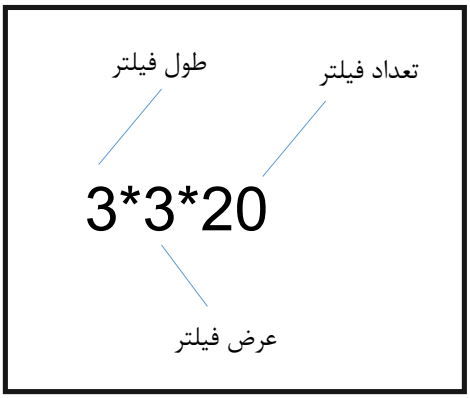
$\max(13,14,18,21) = 21$ Output= **21**

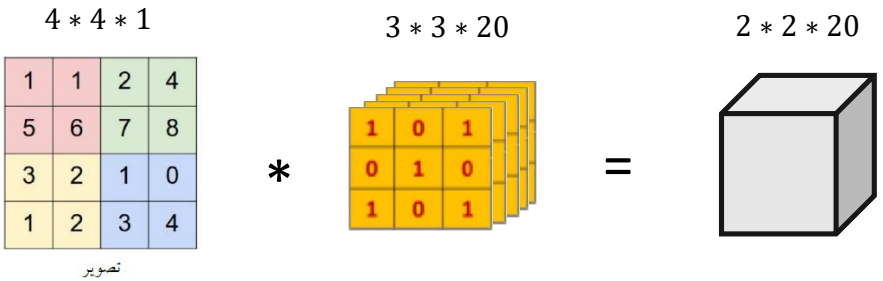
د) در صورتیکه بر روی نتیجه قسمت (الف) و (ب) عملیات mean pooling با سایز ۲ و گام ۲ صورت گیرد، نتیجه چه خواهد بود؟

13	**14**
18	**21**

$\text{mean}(13,14,18,21) = 16.5$ Output= **16.5**

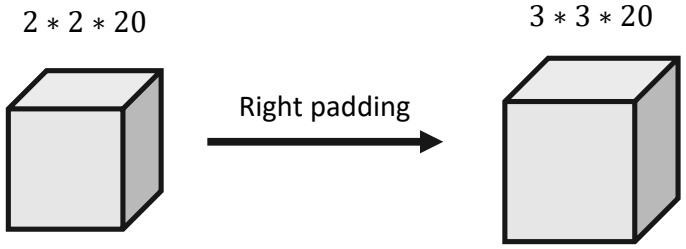
هـ) اگر که یک لایه کانولو شن با سایز ۲۰*۳*۳ بر روی تصویر فوق اعمال شود، در خروجی این لایه چند feature map و با چه اندازه هایی موجود خواهد بود؟

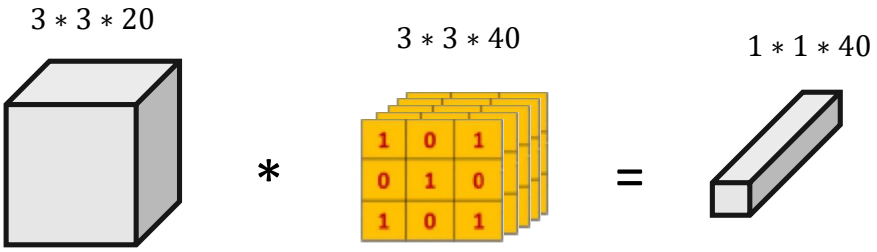




هـ) اگر خروجی لایه کانولو شن ق سمت (ه) به لایه کانولو شن دیگری با سایز ۴۰*۳*۳ داده شود، خروجی این لایه چند feature mapو با چه اندازه هایی است؟

با توجه به اینکه اندازه ی ورودی کوچکتر از اندازه ی فیلتر است، باید از تکنیکی مانند padding استفاده کنیم
توجه شود که عملیات padding می تواند از هر جهتی باشد.





هـ) مفهوم vanishing gradient و exploding gradient را در شبکه های عصبی بازگشتی بیان کنید و توضیح دهید چگونه این مشکل در LSTM حل می شود؟

Vanishing –

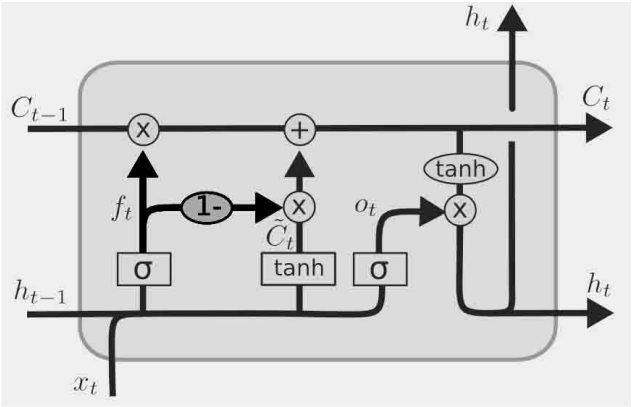
As the backpropagation algorithm advances downwards(or backward) from the output layer towards the input layer, the gradients often get smaller and smaller and approach zero which eventually leaves the weights of the initial or lower layers nearly unchanged. As a result, the gradient descent never converges to the optimum. This is known as the ***vanishing gradients*** problem.

Exploding –

On the contrary, in some cases, the gradients keep on getting larger and larger as the backpropagation algorithm progresses. This, in turn, causes very large weight updates and causes the gradient descent to diverge. This is known as the ***exploding gradients*** problem.

LSTM leverages gating mechanisms to control the flow of information and gradients. This helps prevent the vanishing gradient problem and allows the network to learn and retain information over longer sequences

۳) شکل زیر نشانگر تغییری است که در نورون استاندارد LSTMداده شده است. روابط برای این نورون را بنویسید و شرح دهید به نظر شما این تغییر در ساختار سه بخشی نورون LSTMچه مفهومی دارد و در کجا کاربرد دارد؟ $1-$ در شکل به صورت $1-f_t$ عمل می کند.



$$f_t = \sigma\left(W_i[h_{t-1}, x_t] + b_f\right)$$

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

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

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

$$h_t = o_t * \tanh C_t$$

در این ساختار عکس خروجی گیت فراموشی تعیین می کند که چه المان هایی شامل تغییرات شوند. به عبارت دیگر المان هایی که شانس کمتری از ماندگاری در گام های بعدی از ورودی دارند احتمال تغییر آن ها در گیت ورودی بیشتر خواهد بود.

۴) تفاوت شبکه MLPو DBN از دو نقطه نظر ساختار و کاربرد در چیست؟

A deep belief network is a type of deep learning model that is typically used for unsupervised learning tasks, such as feature learning and dimensionality reduction. It is composed of multiple layers of latent variables, and it uses a restricted Boltzmann machine (RBM) to pre-train each layer before fine-tuning the entire network with backpropagation

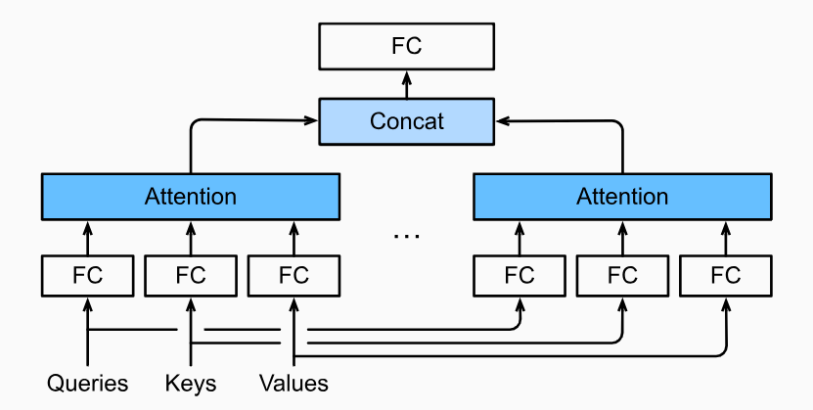
۵) مزیت ترنسفورمر نسبت به LSTMدر کاربردهای پردازش زبان طبیعی چیست؟

While RNNs and LSTMs were the go-to choices for sequential tasks, Transformers have proven to be a viable alternative due to their parallel processing capability, ability to capture long-range dependencies, and improved hardware utilization. the choice between LSTM and Transformer models ultimately depends on the specific requirements of the task at hand, striking a balance between efficiency, accuracy, and interpretability.

۶) در ورودی یک ترنسفورمر، یک جمله ۴ کلمه ای داده شده است که هر کلمه با یک بردار ویژگی ۵۱۲ بعدی توصیف میشود. اگر ماتریسهای key,query,value دارای بعد ۶۴ * ۵۱۲ باشند، مشخص کنید در هر یک از دو حالت زیر خروجی بلوک self-attention و خروجی بخش multi-head attention چند بعدی است.

$$Q = X * W_q : (4,512) * (512,64) : (4,64)$$
$$K = X * W_k : (4,512) * (512,64) : (4,64)$$
$$V = X * W_v : (4,512) * (512,64) : (4,64)$$
$$Z = sfotmax\left(\frac{Q * K^T}{\sqrt{d_k}}\right)V : \frac{(4,64) * (64,4)}{(1,1)} * (4,64) : \frac{(4,64) * (64,4)}{(1,1)} * (4,64) : (4,4) * (4,64) : (4,64)$$

اندازه خروجی ماژول self-attention برابر (4,64) می باشد. در حالتی که از MHA با تعداد هد چهار استفاده می کنیم، یعنی چهار ماژول self attention به طور موازی استفاده می کنیم که خروجی آن ها در نهایت concat می شود. و در حالتی که از MHA با تعداد هد برابر با هشت استفاده می کنیم، یعنی هشت ماژول self attention به طور موازی استفاده می کنیم که خروجی آن ها در نهایت concat می شود. پس اندازه خروجی self-attention در هر دو حالت یکسان است.



4 head : $Z_t = concat(z_1, z_2, z_3, z_4\ ) : concat\big((4 * 64)_1\ , (4 * 64)_2, (4 * 64)_3\ , (4 * 64)_4\big)$: (4,256)

8 head : $Z_t = concat(z_1, z_2, \dots, z_8\ ) : concat\big((4 * 64)_1\ , \dots\ , (4 * 64)_8\big)$: $(4 * 512)$

۷) اگر جمله زیر را در ورودی ترنسفورمر با بردارهای ویژگی نشان داده شده داشته باشیم و بردارهای key,query,value دارای بعد ۳ * ۱ باشد و بصورت تصادفی مقدار دهی شده باشند، با در نظر گرفتن مراحل محاسبه بصورت پارامتری نشان دهید که خروجی بلوک self-attentionچه خواهد شد؟

$X =$

The	0.21	-0.15	0.32
Cat	0.85	0.29	-0.61
Sat	-0.37	0.72	0.45
On	0.12	-0.64	0.27
The	0.21	-0.15	0.32
Mat	-0.53	0.31	0.81

$W_q =$ | q_1 | q_2 | q_3 |

$W_k =$ | k_1 | k_2 | k_3 |

$W_v =$ | v_1 | v_2 | v_3 |

$K = XW_k^T =$

$0.21k_1 - 0.15k_2 + 0.32k_3$
$0.85k_1 + 0.29k_2 - 0.61k_3$
$-0.37k_1 + 0.72k_2 + 0.45k_3$
$0.12k_1 - 0.64k_2 + 0.27k_3$
$0.21k_1 - 0.15k_2 + 0.32k_3$
$-0.53k_1 + 0.31k_2 + 0.81k_3$

ماتریس های V و Q نیز مطابق ماتریس محاسبه شده برای k تشکیل می شوند.

$Q^T =$ | q_1 | q_2 | q_3 | q_4 | q_5 | q_6 |

$K^T =$ | k_1 | k_2 | k_3 | k_4 | k_5 | k_6 |

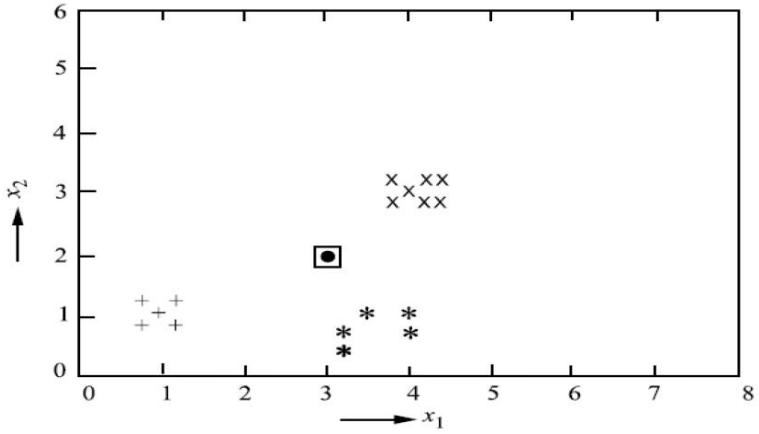
$V^T =$ | v_1 | v_2 | v_3 | v_4 | v_5 | v_6 |

$$Z = sfotmax\left(\frac{Q * K^T}{\sqrt{d_k}}\right)V = softmax\left(\frac{[q_1\ q_2\ q_3\ q_4\ q_5\ q_6] * [k_1\ k_2\ k_3\ k_4\ k_5\ k_6]^T}{\sqrt{3}}\right) * V$$

$$Z = sfotmax\left(\frac{Q * K^T}{\sqrt{d_k}}\right)V = softmax\left(\frac{\begin{bmatrix} q_1k_1 & \cdots & q_1k_6 \\ \vdots & \ddots & \vdots \\ q_6k_1 & \cdots & q_6k_6 \end{bmatrix}}{\sqrt{3}}\right) * V$$

۸) دادگان صفحه بعد را در نظر بگیرید. در هر داده عنصر اول و دوم مختصات و عنصر سوم برچسب یا شماره کلاس است. این داده ها در شکل نیز نشان داده شده اند. می خواهیم با استفاده از روش KNN کلاس نمونه آزمایش (۳,۲) (که در شکل با علامت دایره مشخص شده است) را تعیین کنیم. کلاس این نمونه را در سه حالت k=1,k=3, k=5 مشخص نمایید. از فاصله اقلیدسی برای تعیین نزدیکترین همسایه استفاده کنید.

$X_1 = (0.8, 0.8, 1)$, $X_2 = (1.0, 1.0, 1)$, $X_3 = (1.2, 0.8, 1)$
$X_4 = (0.8, 1.2, 1)$, $X_5 = (1.2, 1.2, 1)$, $X_6 = (4.0, 3.0, 2)$
$X_7 = (3.8, 2.8, 2)$, $X_8 = (4.2, 2.8, 2)$, $X_9 = (3.8, 3.2, 2)$
$X_{10} = (4.2, 3.2, 2)$, $X_{11} = (4.4, 2.8, 2)$, $X_{12} = (4.4, 3.2, 2)$
$X_{13} = (3.2, 0.4, 3)$, $X_{14} = (3.2, 0.7, 3)$, $X_{15} = (3.8, 0.5, 3)$
$X_{16} = (3.5, 1.0, 3)$, $X_{17} = (4.0, 1.0, 3)$, $X_{18} = (4.0, 0.7, 3)$



- فاصله نمونه تست را نسبت به همه ی نمونه های آموزشی محاسبه می کنیم

$$d_1 = d(x_{test}, x_1) = |x_{test} - x_1|_2 = \sqrt{(3-0.8)^2 + (2-0.8)^2} = 2.5$$
$$d_2 = d(x_{test}, x_2) = |x_{test} - x_2|_2 = \sqrt{(3-1)^2 + (2-1)^2} = 2.23$$
$$d_3 = d(x_{test}, x_3) = |x_{test} - x_3|_2 = \sqrt{(3-1.2)^2 + (2-0.8)^2} = 2.16$$
$$d_4 = d(x_{test}, x_4) = |x_{test} - x_4|_2 = \sqrt{(3-0.8)^2 + (2-1.2)^2} = 2.34$$
$$d_5 = d(x_{test}, x_5) = |x_{test} - x_5|_2 = \sqrt{(3-1.2)^2 + (2-1.2)^2} = 1.96$$
$$d_6 = d(x_{test}, x_6) = |x_{test} - x_6|_2 = \sqrt{(3-4)^2 + (2-3)^2} = 1.41$$
$$d_7 = d(x_{test}, x_7) = |x_{test} - x_7|_2 = \sqrt{(3-3.8)^2 + (2-2.8)^2} = 1.13$$
$$d_8 = d(x_{test}, x_8) = |x_{test} - x_8|_2 = \sqrt{(3-4.2)^2 + (2-2.8)^2} = 1.44$$
$$d_9 = d(x_{test}, x_9) = |x_{test} - x_9|_2 = \sqrt{(3-3.8)^2 + (2-3.2)^2} = 1.44$$
$$d_{10} = d(x_{test}, x_{10}) = |x_{test} - x_{10}|_2 = \sqrt{(3-4.2)^2 + (2-3.2)^2} = 1.69$$
$$d_{11} = d(x_{test}, x_{11}) = |x_{test} - x_{11}|_2 = \sqrt{(3-4.4)^2 + (2-2.8)^2} = 1.61$$
$$d_{12} = d(x_{test}, x_{12}) = |x_{test} - x_{12}|_2 = \sqrt{(3-4.4)^2 + (2-3.2)^2} = 1.84$$
$$d_{13} = d(x_{test}, x_{13}) = |x_{test} - x_{13}|_2 = \sqrt{(3-3.2)^2 + (2-0.4)^2} = 1.61$$
$$d_{14} = d(x_{test}, x_{14}) = |x_{test} - x_{14}|_2 = \sqrt{(3-3.2)^2 + (2-0.7)^2} = 1.31$$
$$d_{15} = d(x_{test}, x_{15}) = |x_{test} - x_{15}|_2 = \sqrt{(3-3.8)^2 + (2-0.5)^2} = 1.7$$
$$d_{16} = d(x_{test}, x_{16}) = |x_{test} - x_{16}|_2 = \sqrt{(3-3.5)^2 + (2-1)^2} = 1.11$$
$$d_{17} = d(x_{test}, x_{17}) = |x_{test} - x_{17}|_2 = \sqrt{(3-4)^2 + (2-1)^2} = 1.41$$
$$d_{18} = d(x_{test}, x_{18}) = |x_{test} - x_{18}|_2 = \sqrt{(3-4)^2 + (2-0.7)^2} = 1.64$$

- فاصله ها را بر اساس اندازه مرتب می کنیم

$$d_{16}, d_7, d_{14}, d_6, d_{17}, d_8, d_9, d_{11}, d_{13}, d_{18}, d_{10}, d_{15}, d_{12}, d_5, d_3, d_2, d_4, d_1$$

- کلاس مربوط به نمونه ی آموزشی موجود در هر فاصله های مرتب شده را می نویسیم

$$c_3, c_2, c_3, c_2, c_3, c_2, c_2, c_2, c_3, c_3, c_2, c_3, c_2, c_1, c_1, c_1, c_1, c_1$$

- برای $k = 1$ کلاس با کمترین فاصله را به نمونه تست نسبت می دهیم c_3

- برای $k = 3$ سه کلاس اول با کمترین فاصله را در نظر میگیریم c_3,c_2,c_3 . مشاهده می شود بیشترین کلاس موجود کلاس ۳ می باشد. پس در این حالت نمونه ی تست به کلاس سوم برچسب گذاری می شود.

- برای $k = 5$ پنج کلاس اول با کمترین فاصله را در نظر میگیریم c_3,c_2,c_3,c_2,c_3 . مشاهده می شود بیشترین کلاس موجود کلاس ۳ می باشد. پس در این حالت نیز نمونه ی تست به کلاس سوم برچسب گذاری می شود.