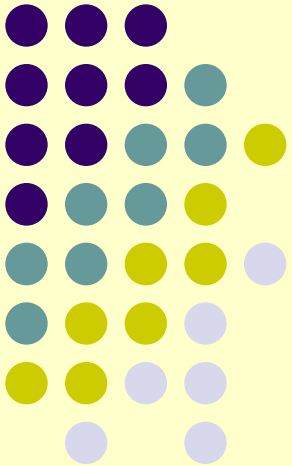
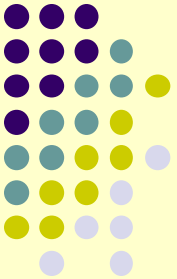
یادگیری عمیق
Deep Learning



استاد درس : محمد امین‌زاده



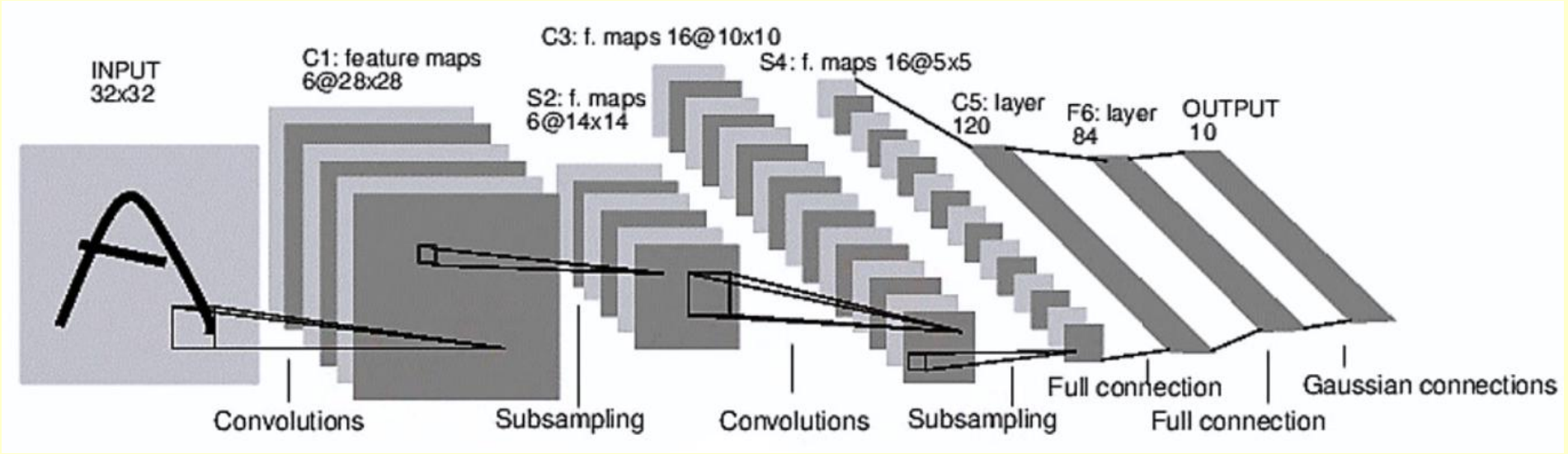
فصل پنجم
شبکه عصبی کانولوشنی (CNN)

معرفی مختصر

- نوع یادگیری بانظارت را استفاده می‌کنند.
- بسیار پرکاربرد و محبوب هستند.
- برخی از کاربردها با شبکه‌های عمیق سنتی قابل پیاده‌سازی نیستند.
- برای یک تصویر ۵۱۲*۵۱۲*۳ در صورت استفاده از ساختار تمام متصل چند نورون ورودی می‌خواهید؟
- ویژگی‌هایی که آن‌ها را از دیگر شبکه‌ها متمایز می‌کند:
  - استفاده از لایه‌های کانولوشنی (conv)
  - چینش نورون‌ها در یک فضای ۳ بعدی
  - لایه کاهش اندازه (Pooling)
  - استفاده از وزن‌های پنجره‌ای (فیلترها)
  - استفاده از چند فیلتر در هر لایه

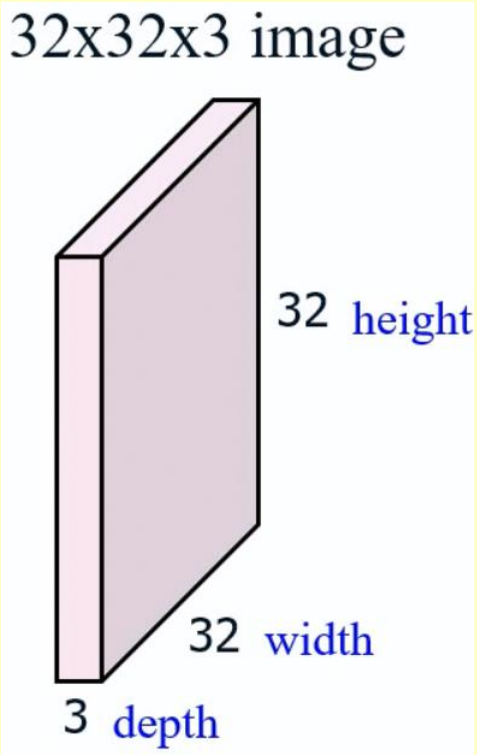
اولین نمونه موفق

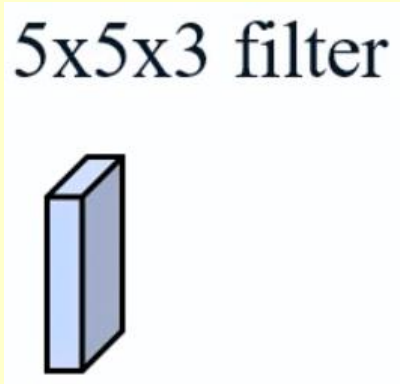
- شبکه LeNet در دهه نود میلادی
- تشخیص اعداد و حروف دستنویس



استفاده از لایه‌های کانولوشنی (conv)

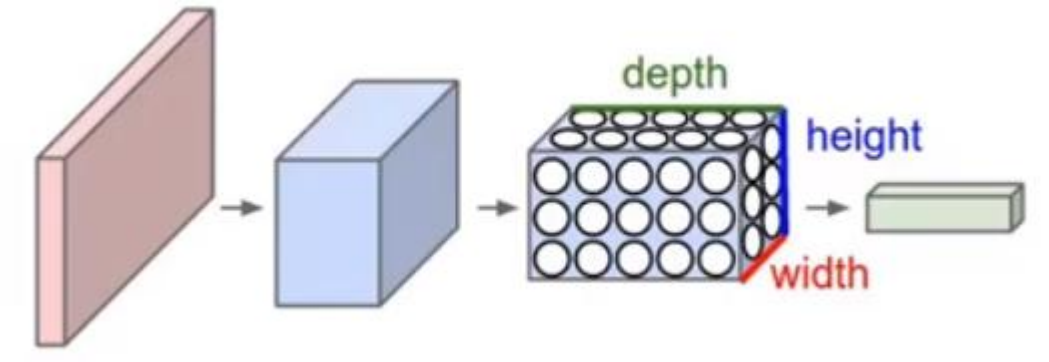
- قسمتی از نورون‌های یک لایه به یک نورون از لایه بعد متصل است.
- هر نورون مسئول شناسایی بخش محدودی از لایه قبل (ورودی) است.
- در تصاویر بزرگ بسیاری از اتصالات بهره‌ای ندارند و می‌توان آن‌ها را حذف کرد. بین پیکسل ابتدا و انتهای تصویر چه ارتباطی وجود دارد؟
- شما می‌خواهید بخش‌هایی از تصویر که به هم ارتباط دارند توسط یک نورون تحلیل شوند.
- با کاهش اتصالات سرعت آموزش افزایش می‌یابد.
- نیاز به داده کمتری برای آموزش است.





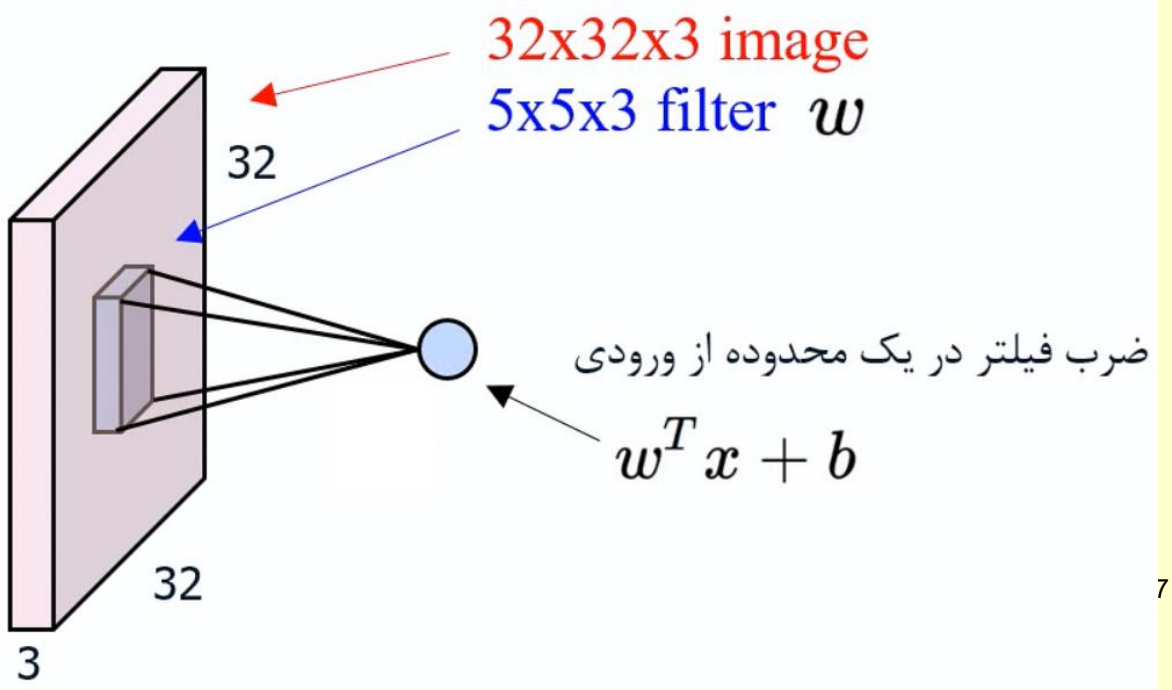
چینش نورون‌ها در یک فضای ۳ بعدی

- نورون‌ها در یک فضای ۳ بعدی چیده شده‌اند.
- چینش در عرض، ارتفاع و عمق است.
- کنار هم بودن نورون‌ها در این ساختار به این مفهوم است که کارهای شبیه به هم انجام می‌دهند.
- منظور از عمق تعداد لایه‌ها نیست.
- این ساختار، سازگار با ورودی تصویر است.
- لایه ورودی متناسب با پیکسل‌های تصویر ورودی است و عمق آن ۳ است.
- عمق بقیه لایه‌ها، متناسب با انتخاب فیلترها خواهد بود.
- در لایه آخر تصویر ورودی به برداری در امتداد عمق کاهش می‌یابد.
- نورون‌های یک لایه با هم اتصال ندارند.

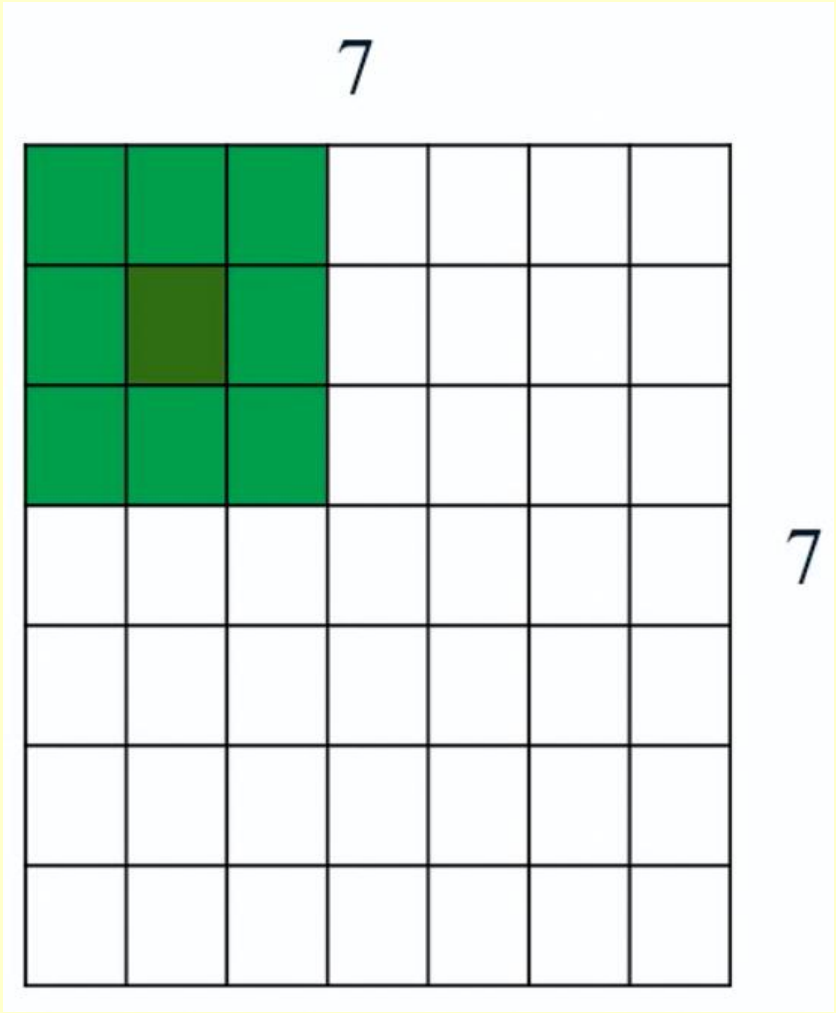


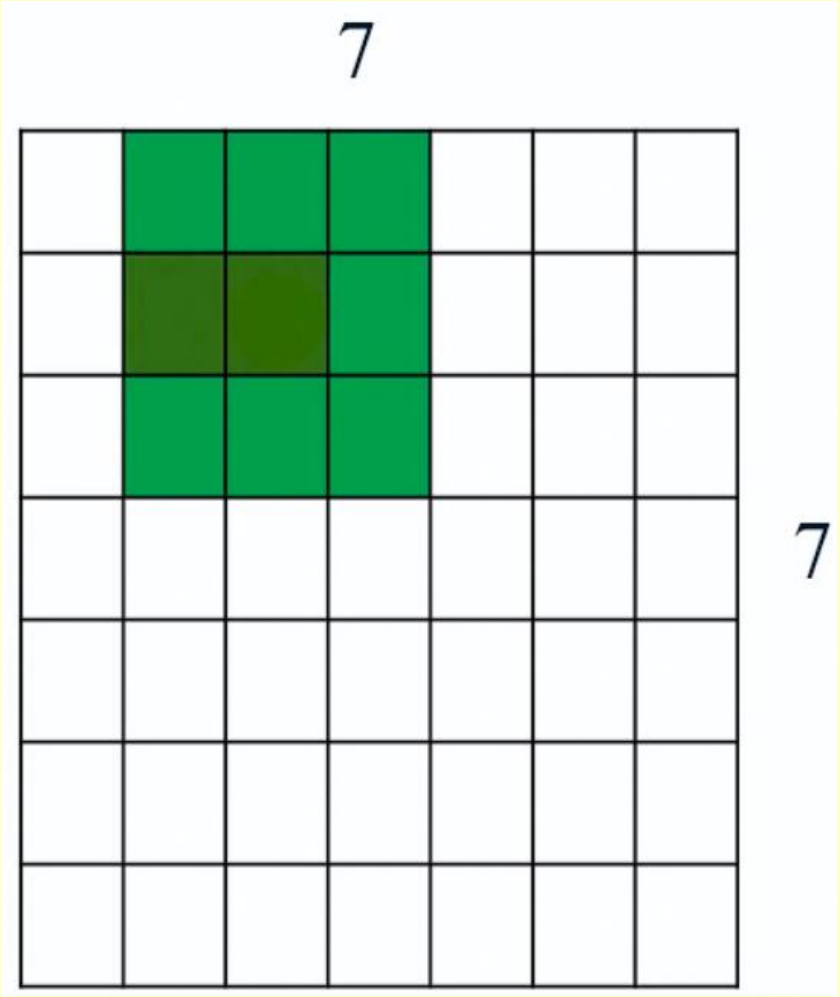
کانولوشن ورودی و فیلتر تصویری

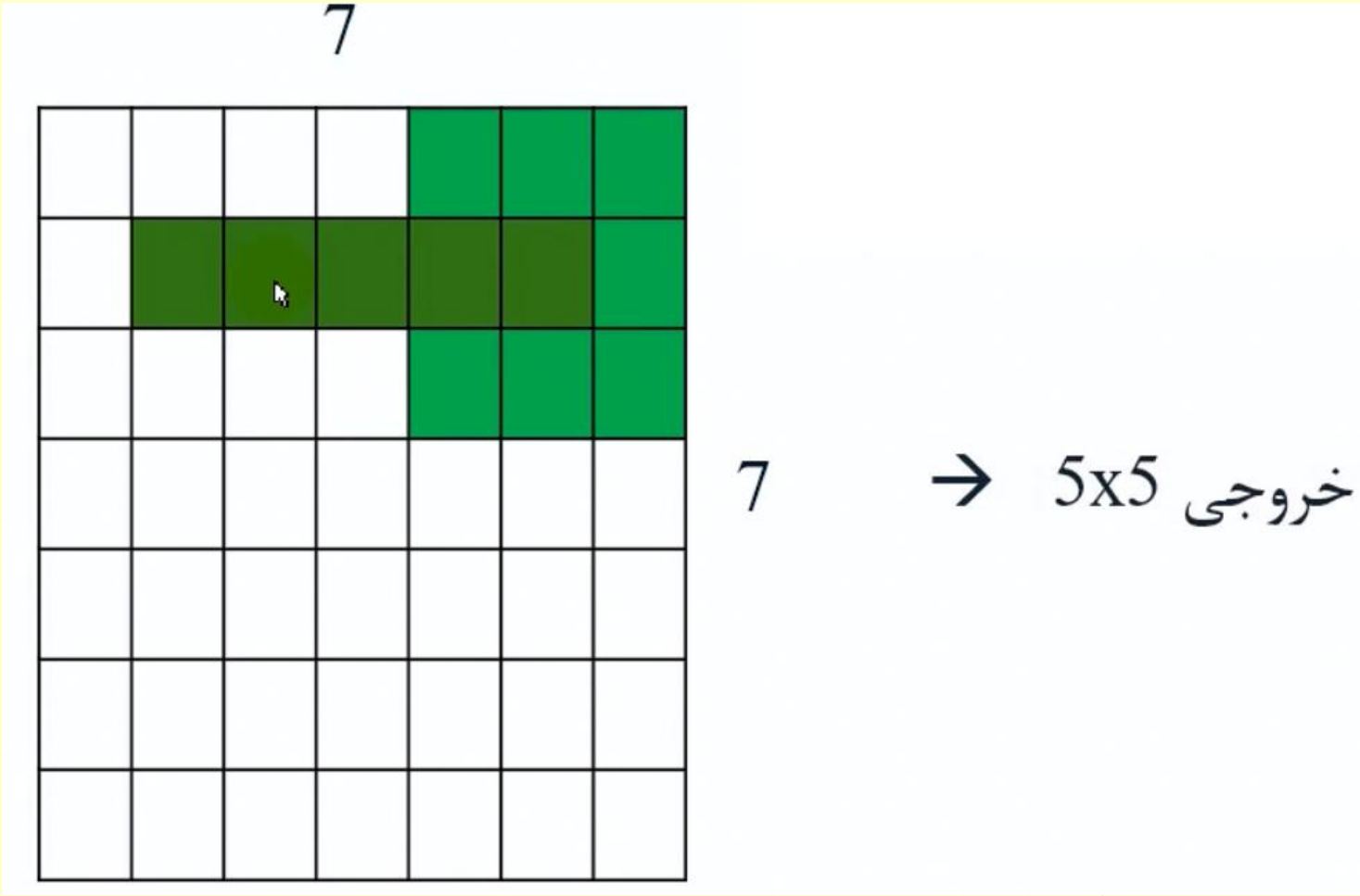
- لغزاندن پنجره بر روی تصویر و محاسبه ضرب نقطه‌ای
- کانولوشن، عملیات وارون کردن ماتریس هسته به صورت افقی و عمودی و سپس ضرب نظیر به نظیر درایه‌های دو ماتریس و در نهایت جمع همه آنهاست. البته اگر ماتریس هسته متقارن باشد نیازی به وارون کردن افقی و عمودی ماتریس نیست زیرا در این حالت هسته وارون شده با هسته اصلی برابر است.



ورودی ۷*۷ و فیلتر ۳*۳



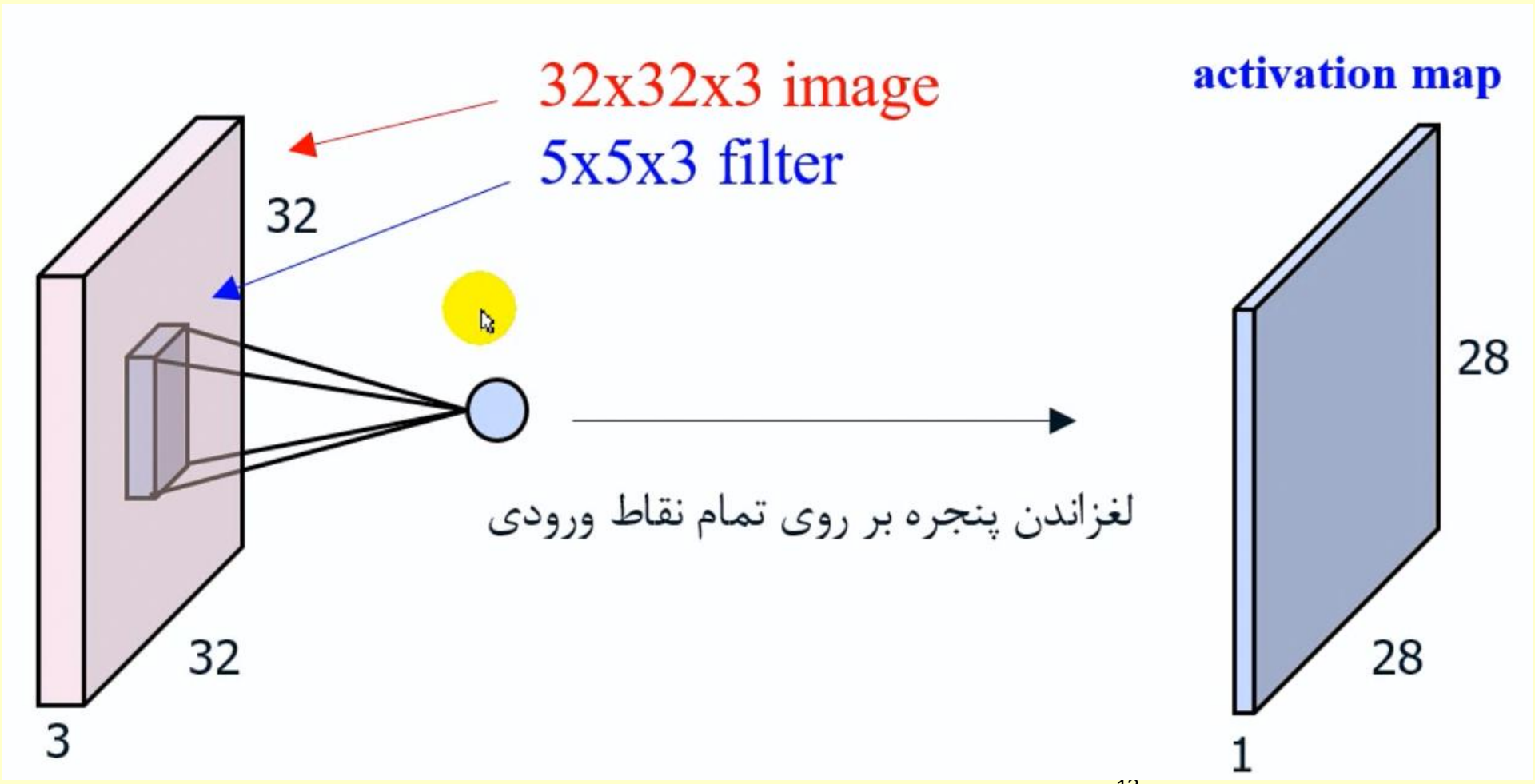



7
7
5x5 خروجی

عملگرها	هسته	تصویر حاصل
همانی	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
تشخیص لبه	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
تیز کردن	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box Blur (نرمال شده)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

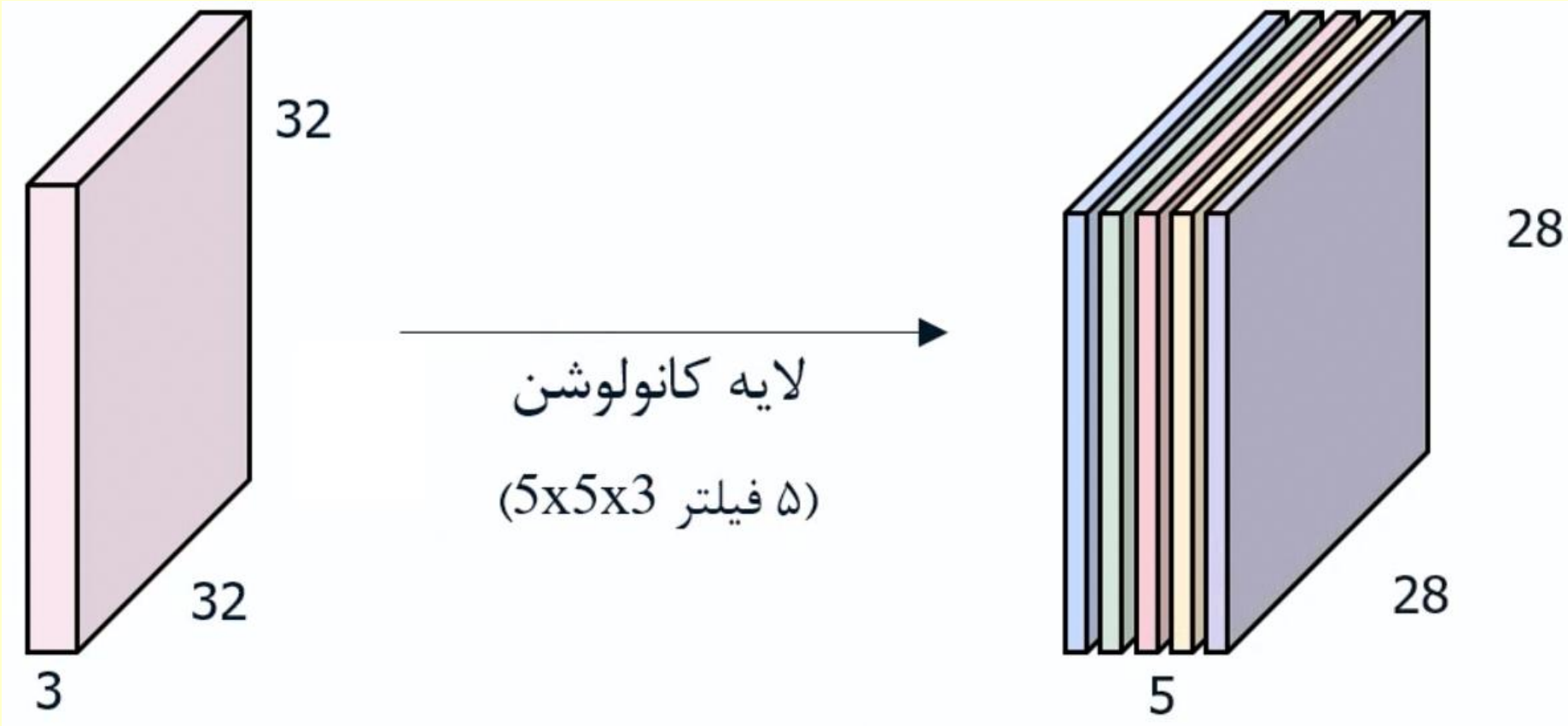
	$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$	**Gaussian Blur 3 × 3** (تقریبی)
	$\frac{1}{256}\begin{bmatrix}1 & 4 & 6 & 4 & 1\\ 4 & 16 & 24 & 16 & 4\\ 6 & 24 & 36 & 24 & 6\\ 4 & 16 & 24 & 16 & 4\\ 1 & 4 & 6 & 4 & 1\end{bmatrix}$	**Gaussian Blur 5 × 5** (تقریبی)
	$\frac{-1}{256}\begin{bmatrix}1 & 4 & 6 & 4 & 1\\ 4 & 16 & 24 & 16 & 4\\ 6 & 24 & -476 & 24 & 6\\ 4 & 16 & 24 & 16 & 4\\ 1 & 4 & 6 & 4 & 1\end{bmatrix}$	**Unsharp masking 5 × 5** بر اساس Gaussian blur مقدار برابر ۱ و آستانه برار • می‌باشد (بدون تصویر ماسک)

خروجی کانولوشن



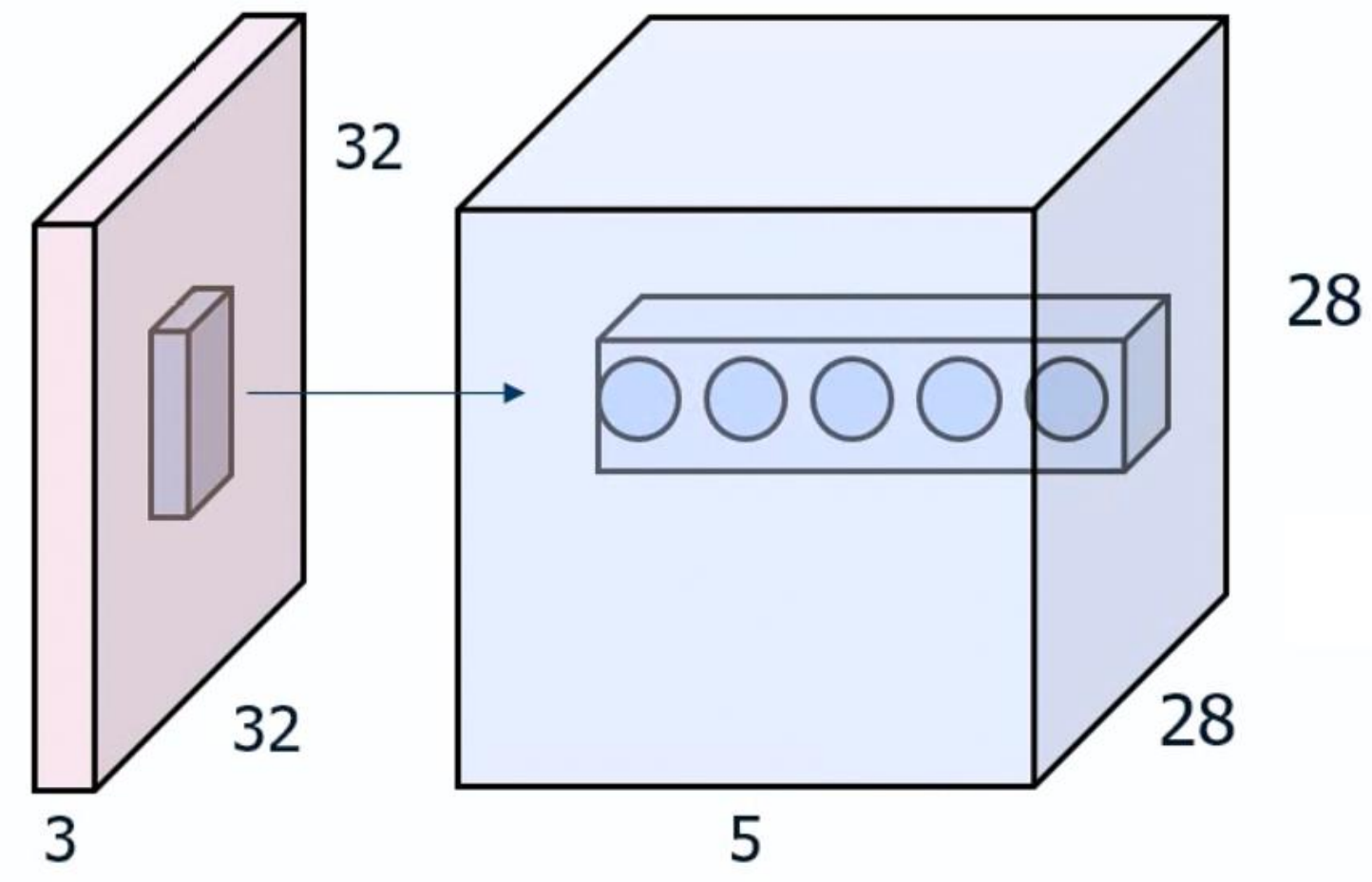
تغییر مقادیر داخلی فیلتر (وزن‌ها) : استفاده از وزن‌های پنجره‌ای

- وزن‌ها به صورت پنجره‌هایی روی ورودی می‌لغزند و تمام ورودی را اسکن می‌کنند.
- با استفاده از این پنجره‌ها هر نورون تنها به نورون‌های داخل پنجره در لایه قبل متصل می‌شود.
- این ویژگی موجب می‌شود که قابلیت ایجاد چند خروجی متفاوت در هر لایه وجود داشته باشد.



شکل دیگری از نمایش

- هر نورون ورودی به ۵ نورون در لایه بعد متصل است. ۵ دیدگاه مختلف از نورون ورودی داریم.

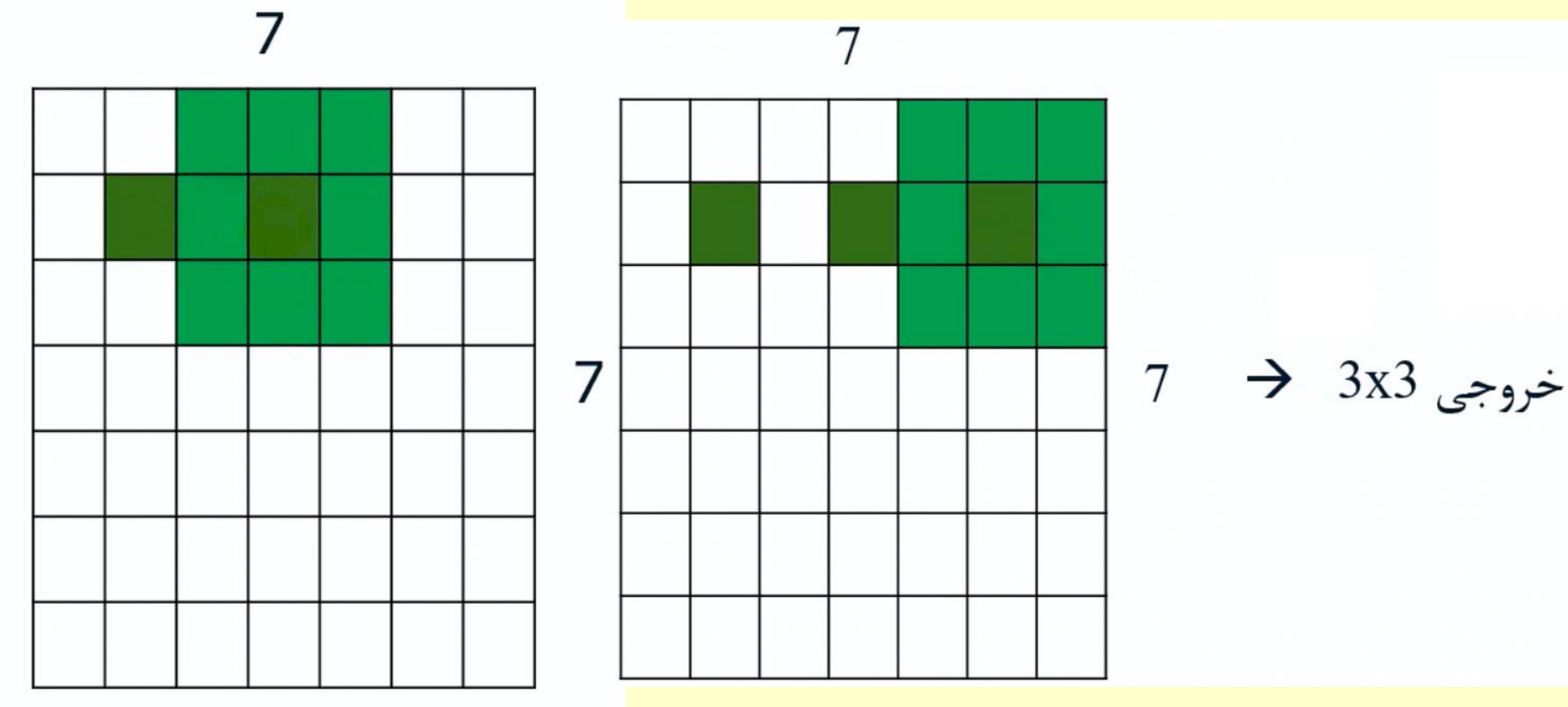


لایه کاهش اندازه (Pooling)

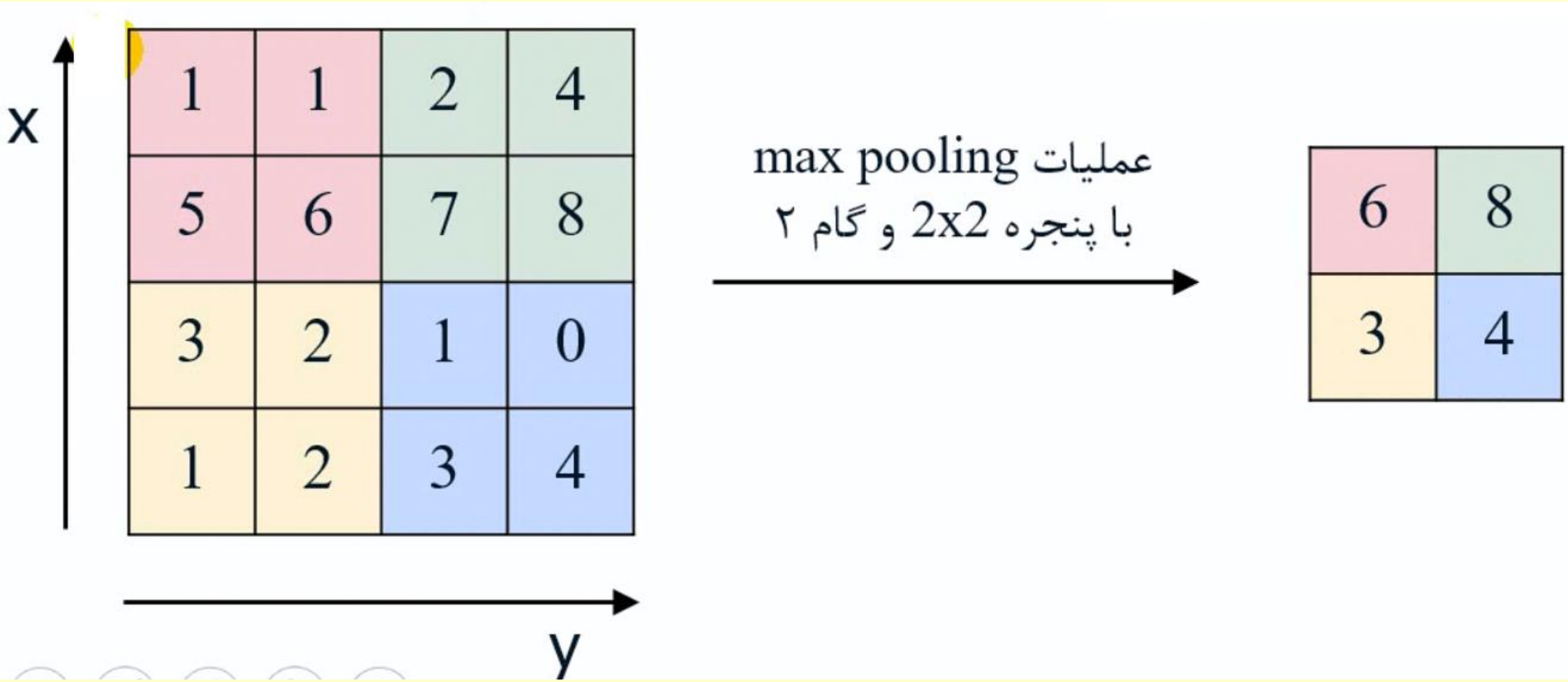
- با افزایش فیلترها، قابلیت شبکه افزایش می‌یابد ولی حجم پردازش سنگین می‌شود و در مواردی آموزش با مشکل مواجه می‌شود.
- استفاده از لایه‌هایی که در آن‌ها فقط کاهش اندازه انجام می‌شود. یعنی ورودی را با اندازه‌ای خاص می‌گیرد و در خروجی، ورودی با اندازه‌ای کمتر شکل می‌گیرد.
- هدف: کاهش اندازه داده‌ها و کاهش حجم محاسبات
- روش‌های کاهش اندازه:
  - استفاده از گام برای پنجره کانولوشن: همراه با کانولوشن عمل کاهش اندازه را انجام می‌دهند.
  - استفاده از لایه Pooling : فقط سایز داده کم می‌شود.

استفاده از گام برای پنجره کانولوشن

- استفاده از گام ۲ برای ورودی ۷*۷ و فیلتر ۳*۳



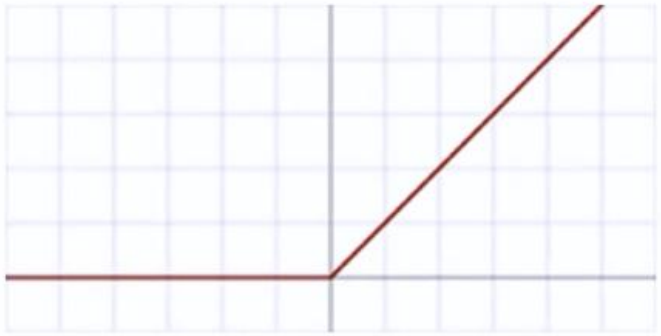
عملیات Max Pooling



تابع غیر خطی

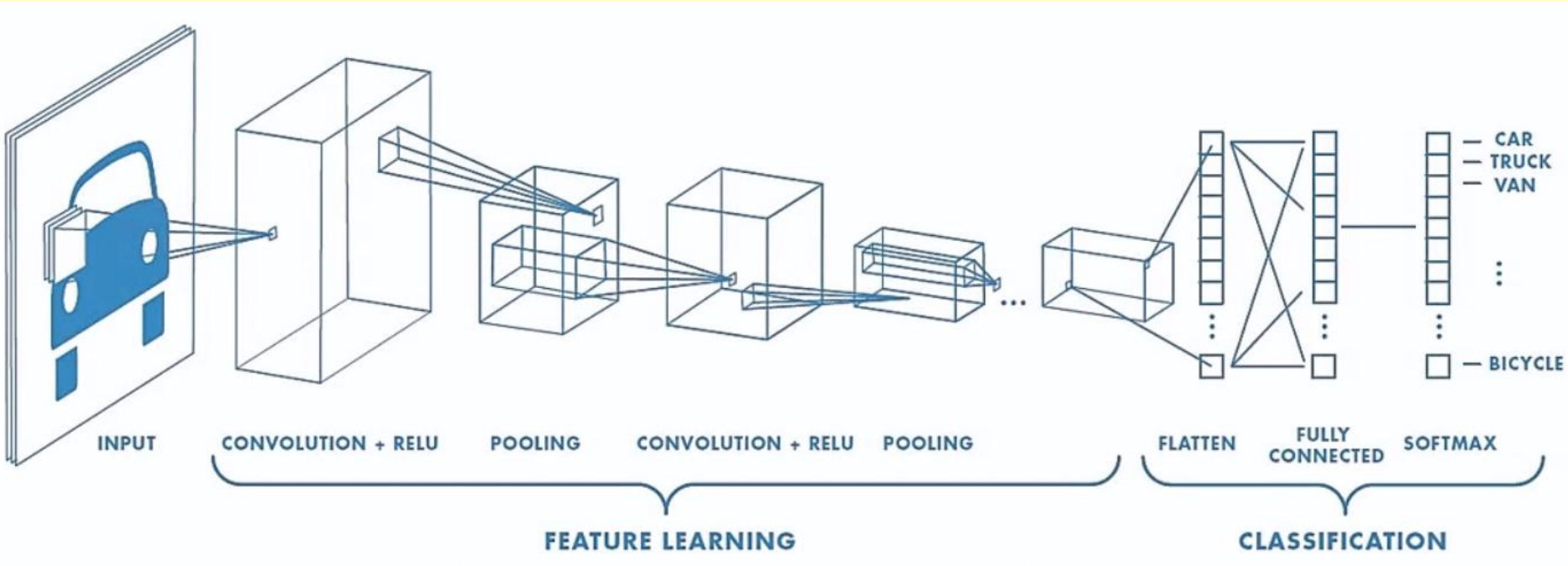
- در این شبکه‌ها معمولا از ReLu استفاده می‌شود.

$$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$$

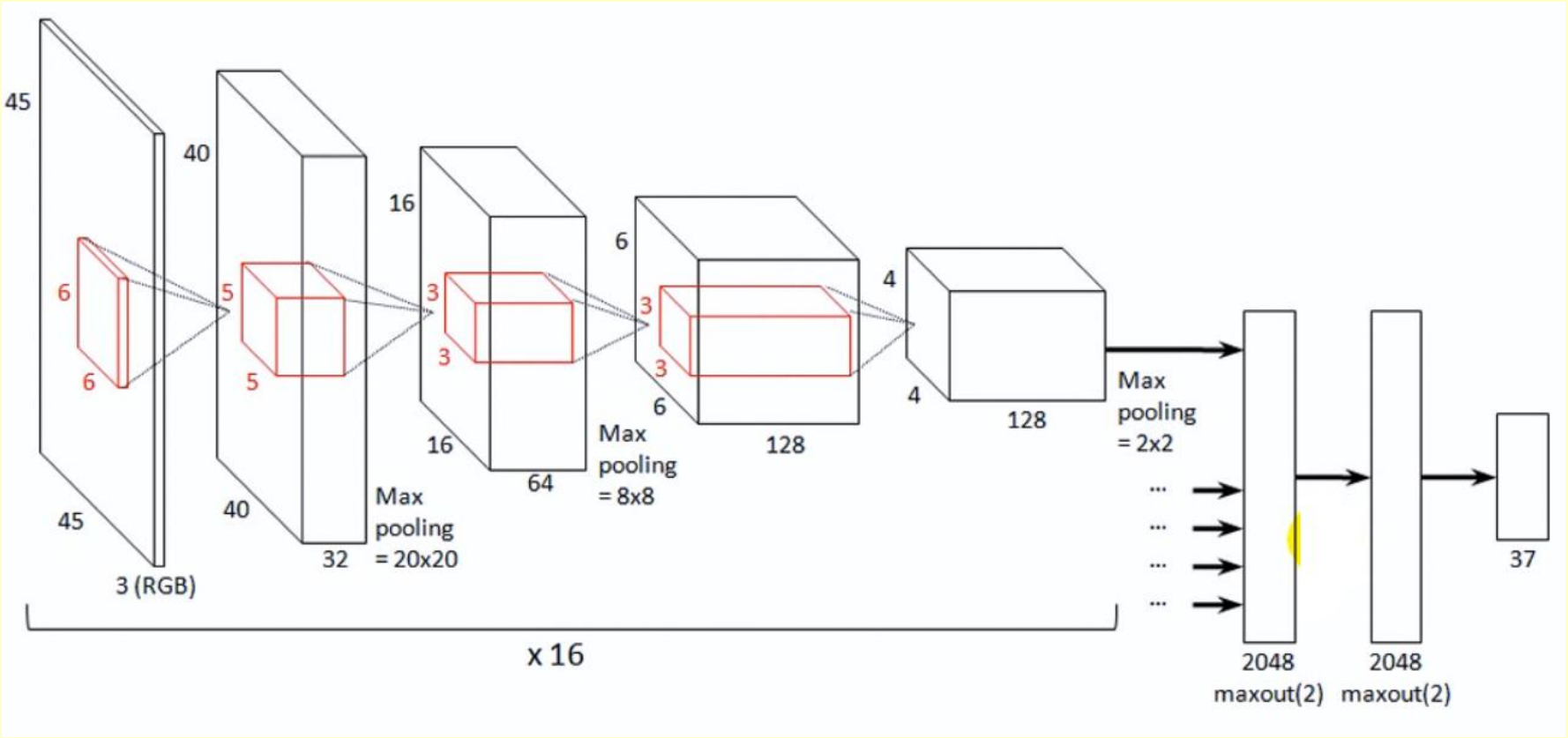


ساختار کلی CNN

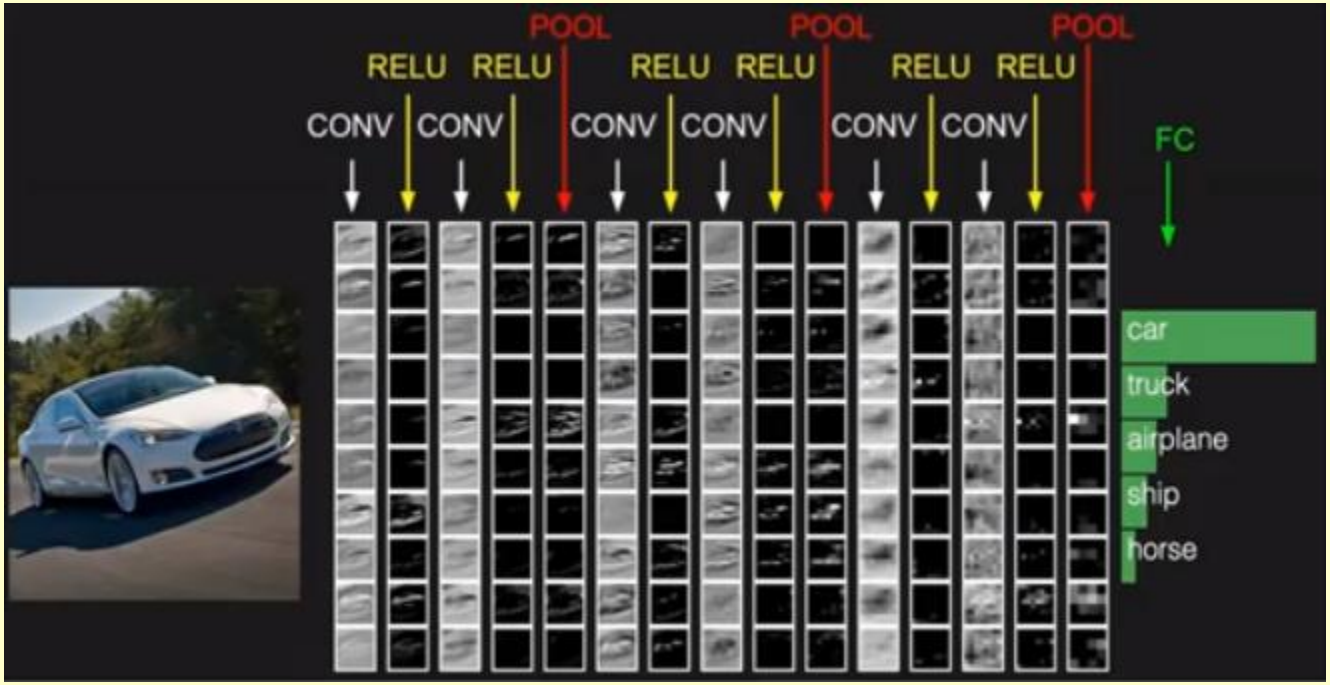
- ورودی داده تصویری
- قسمتی از ورودی به یک نورون از لایه بعد متصل شده است.
- برای دسته‌بندی، نورون‌های تمام متصل در انتها آورده شده است.
- ساختار شبکه از دو بخش یادگیری ویژگی‌ها و دسته‌بندی تشکیل شده است.



نمونه‌ای دیگر از ساختار کلی CNN

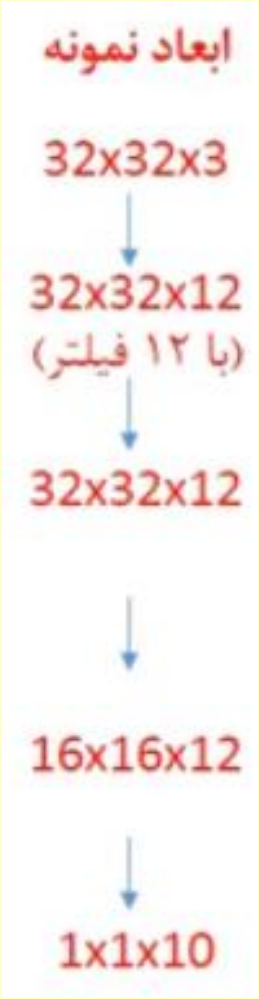


مثال) تشخیص کلاس اشیا

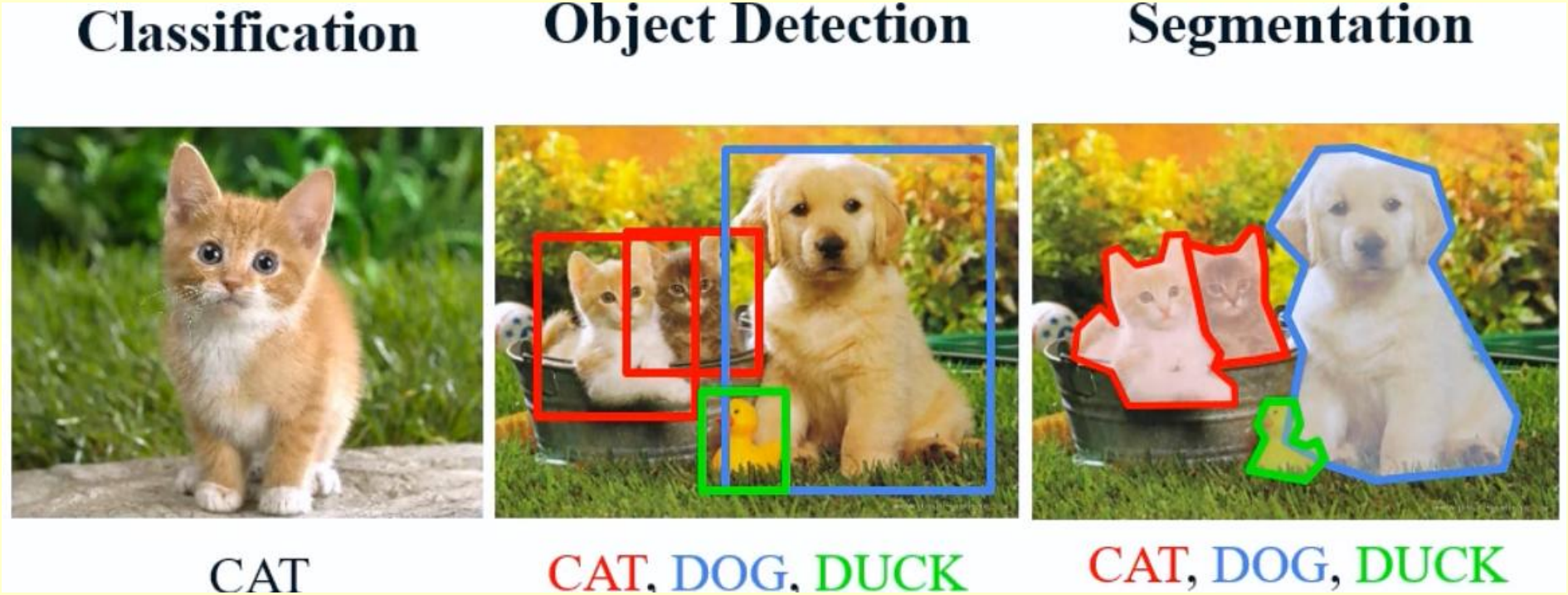


ساختار کلی CNN

- ورودی
  - گرفتن مقادیر خام از نقاط ورودی
- کانولوشن
  - محاسبه ضرب نقطه‌ای بین وزن‌های نورون و یک ناحیه کوچک از حجم ورودی
- اصلاح خطی (RELU)
  - اعمال یک تابع به عناصر
  - عدم تغییر ابعاد حجم
- POOL
  - کاهش نمونه‌برداری در امتداد ابعاد مکانی (عرض و ارتفاع)
- FC
  - محاسبه امتیاز هر کلاس



کاربردهای شبکه CNN

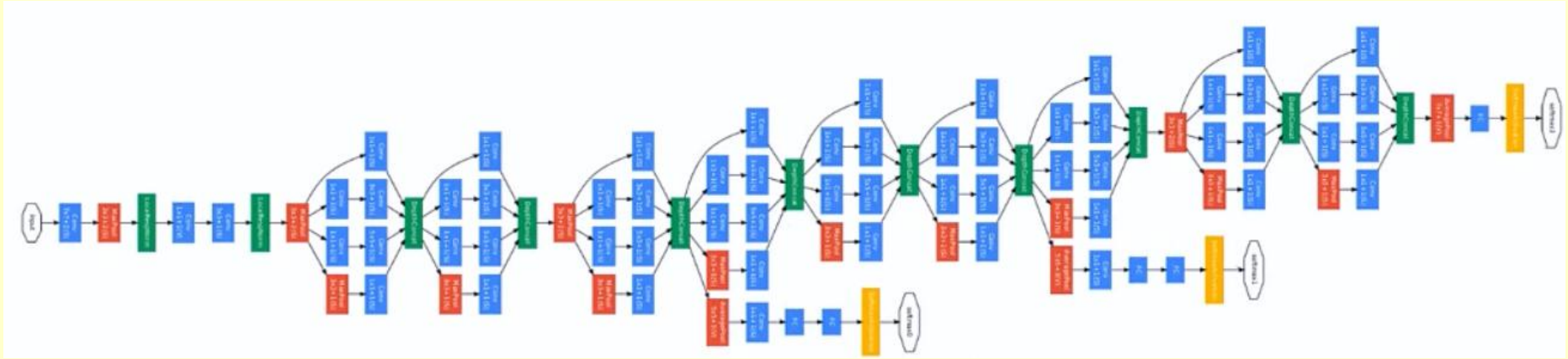


معماری‌های مختلف CNN

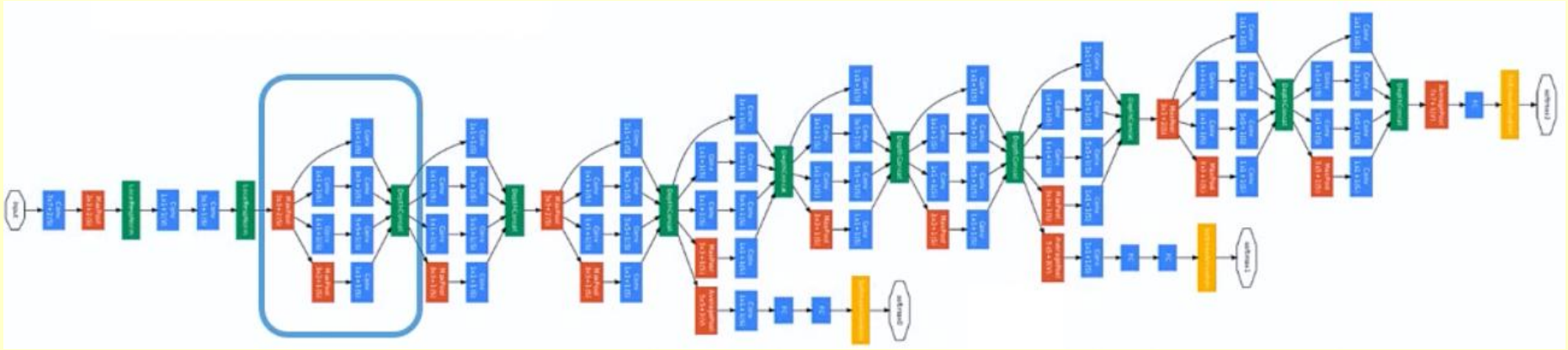
- GoogLeNet
- ResNet
- U-Net
- RCNN

مدل GoogLeNet

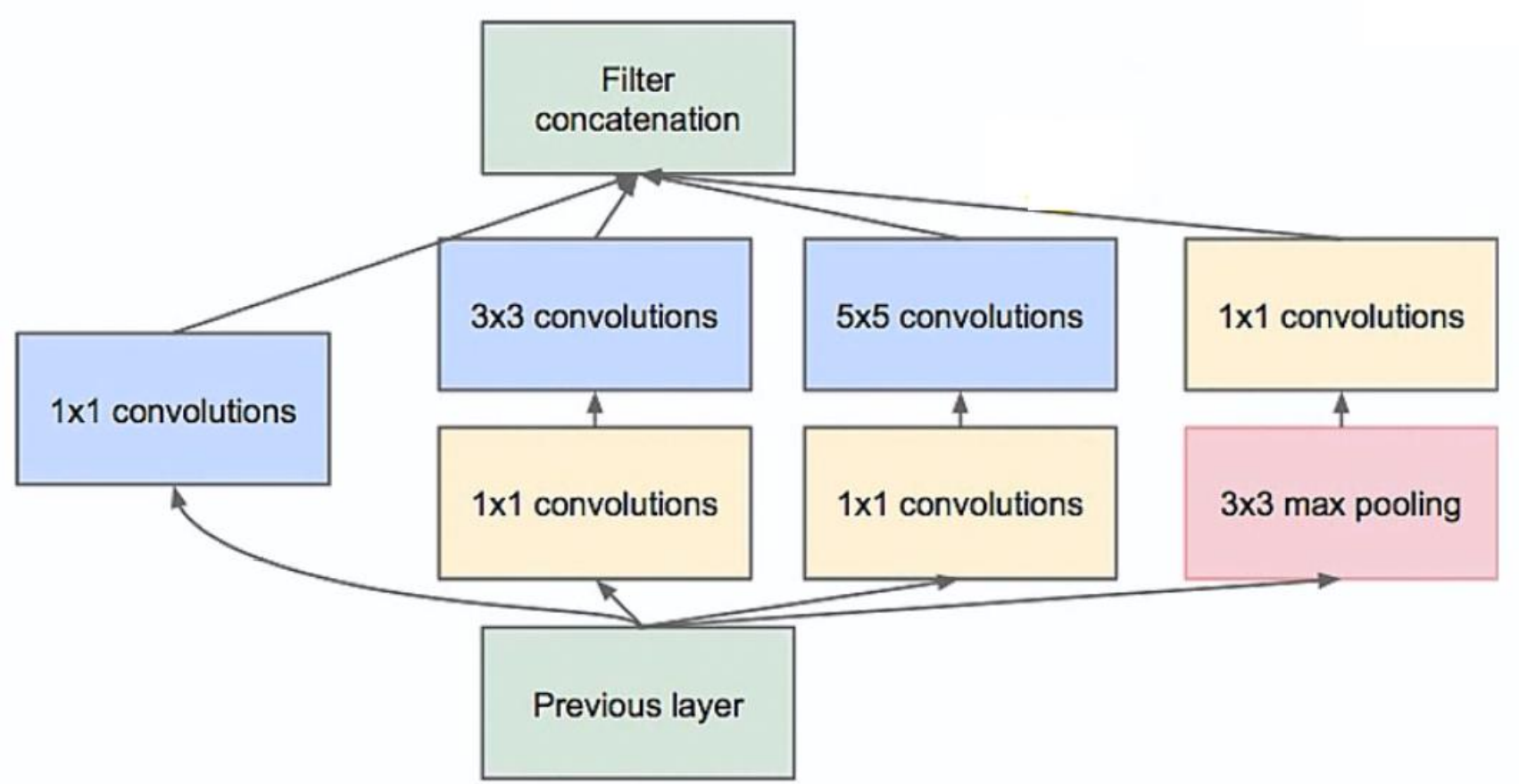
- توسط شرکت گوگل طراحی شده است.



- استفاده از ماژول Inception

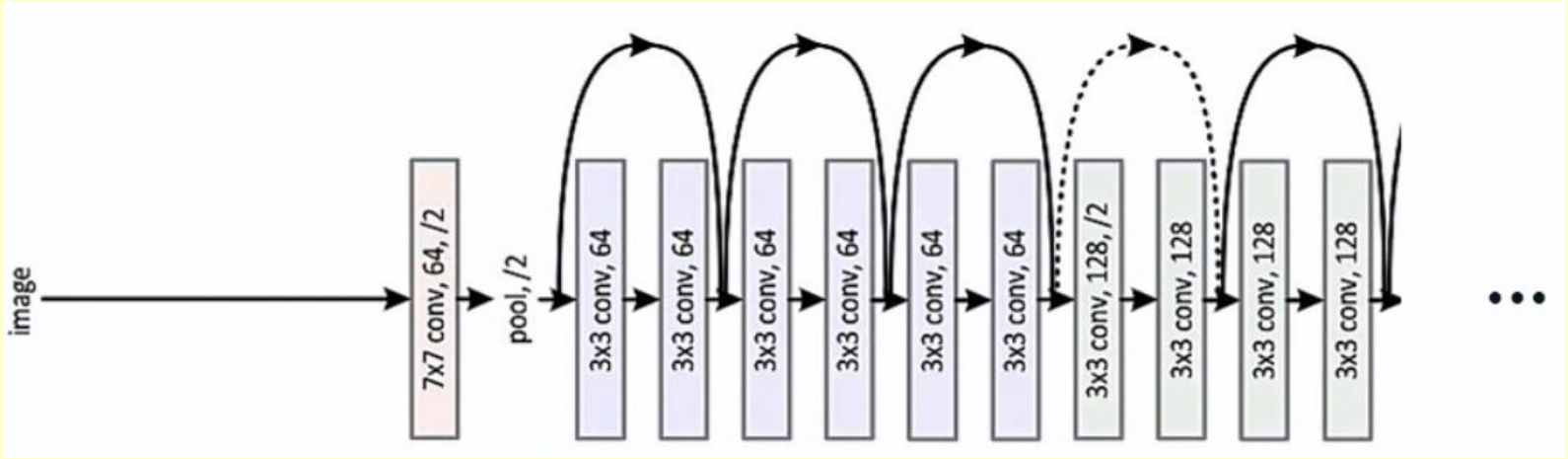


ماژول Inception

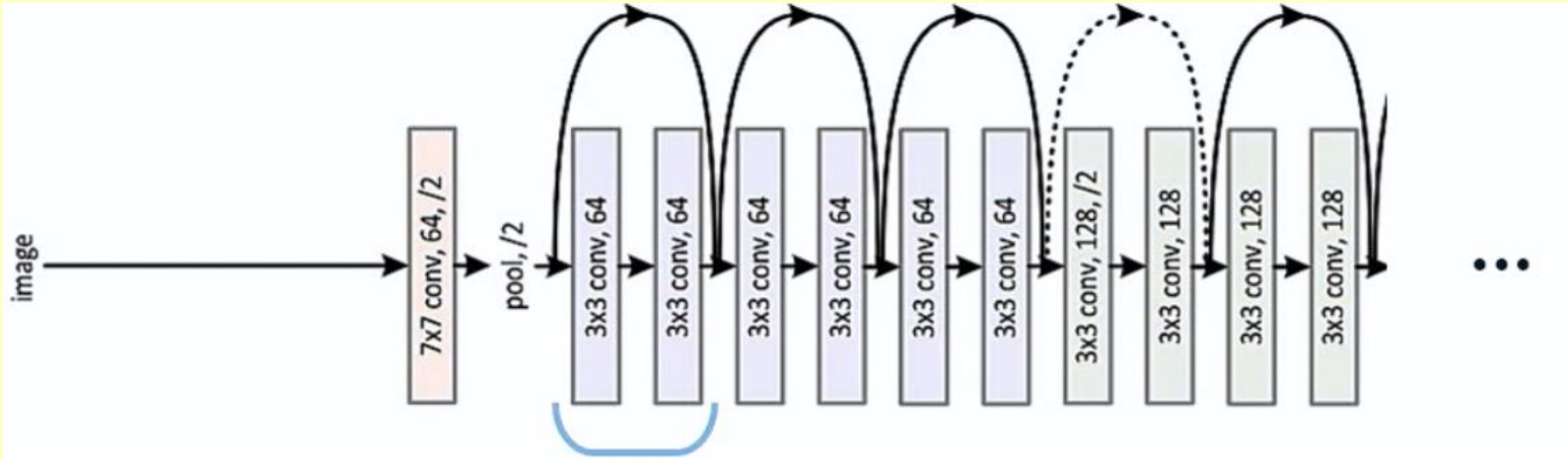


مدل ResNet

- قسمتی از شبکه ResNet (بیش از ۱۰۰ لایه)

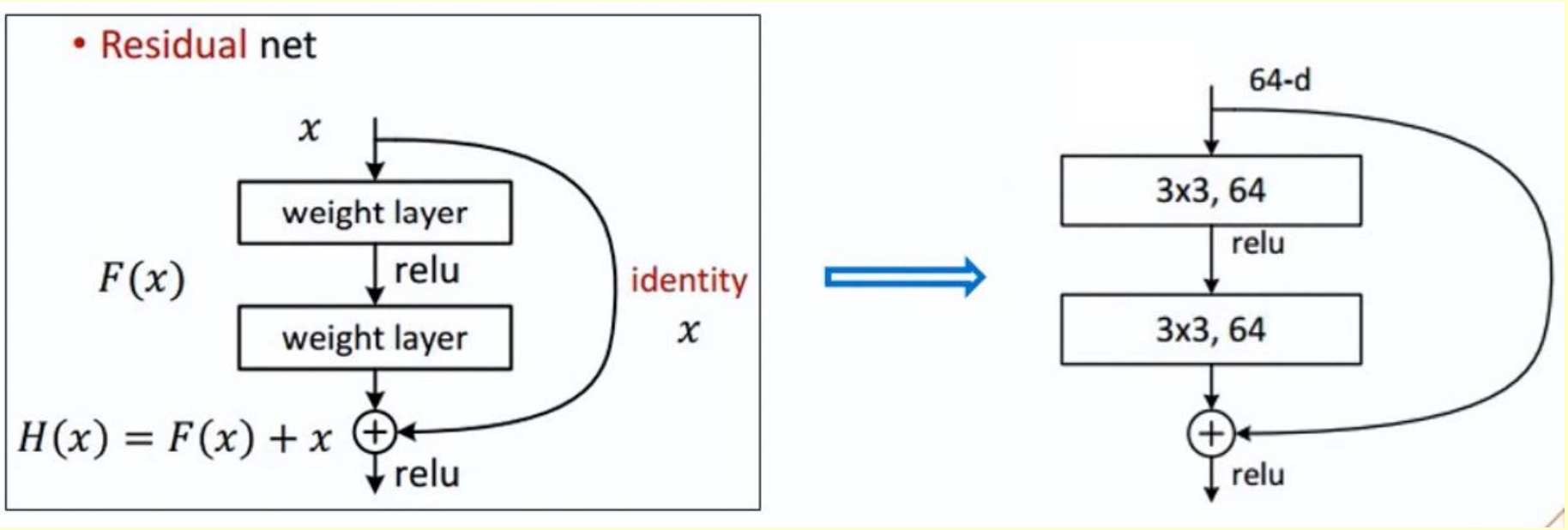


- ترکیب بلوک‌های Residual



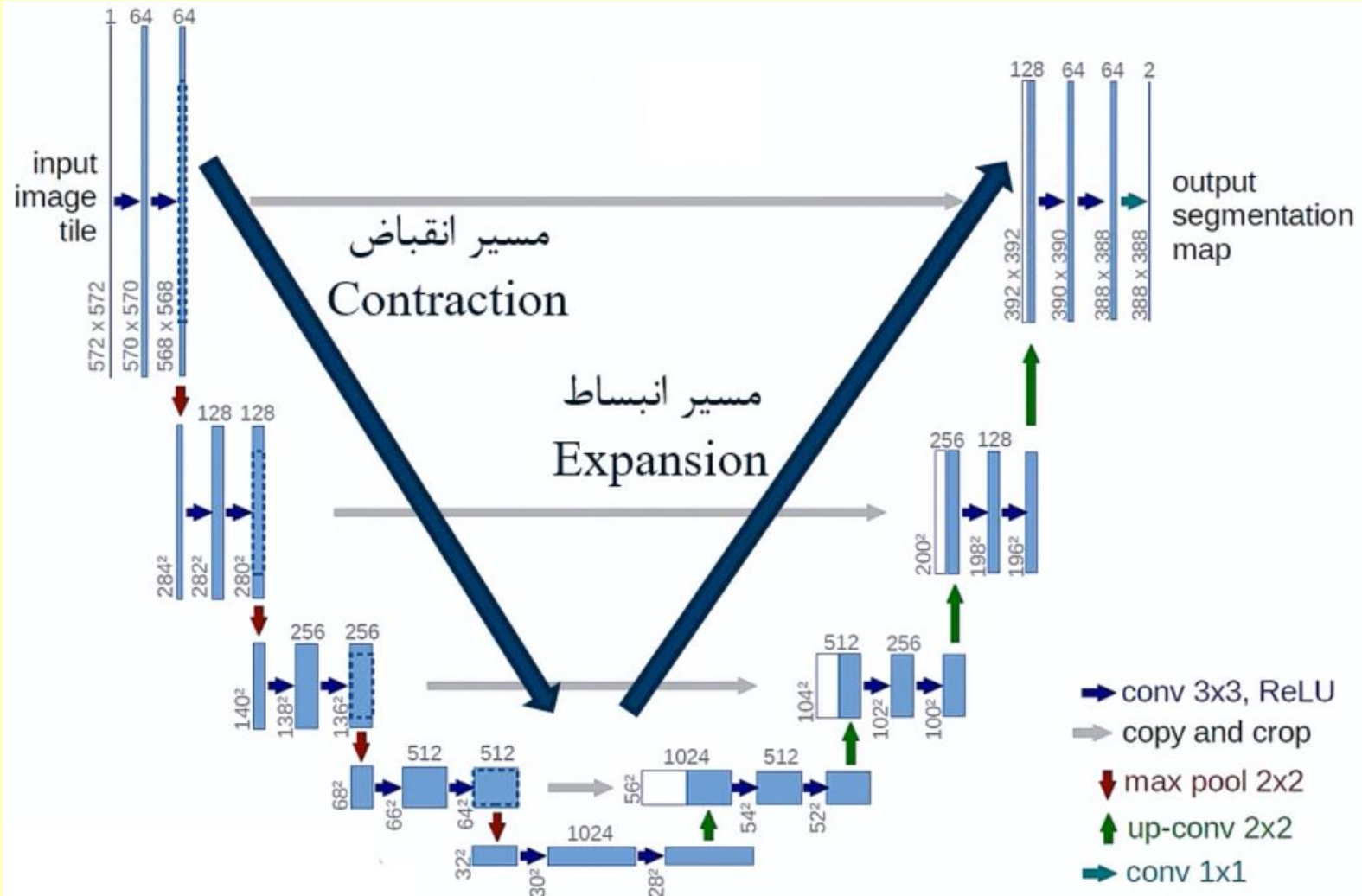
یک نمونه بلوک Residual

- جمع شدن ورودی با خروجی کانوال موجب می‌شود که اطلاعات از دست رفته در فرآیند کانوال دوباره برگردند.
- این عملیات می‌تواند بارها تکرار شود و نگران از دست دادن جزئیات نیستیم.
- اندازه فیلتر، تعداد فیلتر و گام پارامترهای مهم هر لایه هستند.



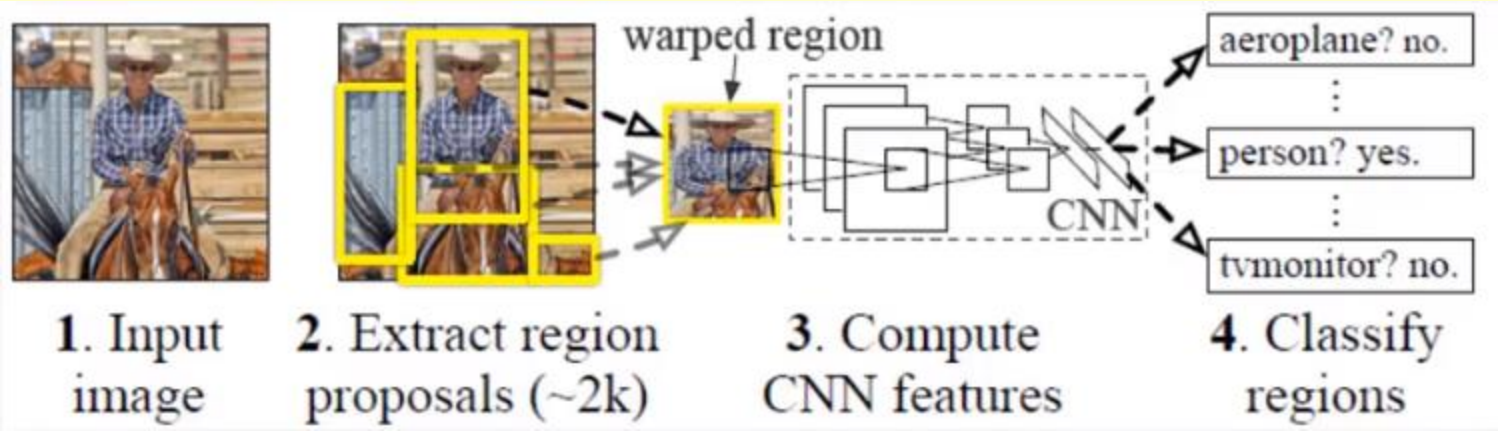
مدل U-Net

- برای بخش‌بندی تصاویر



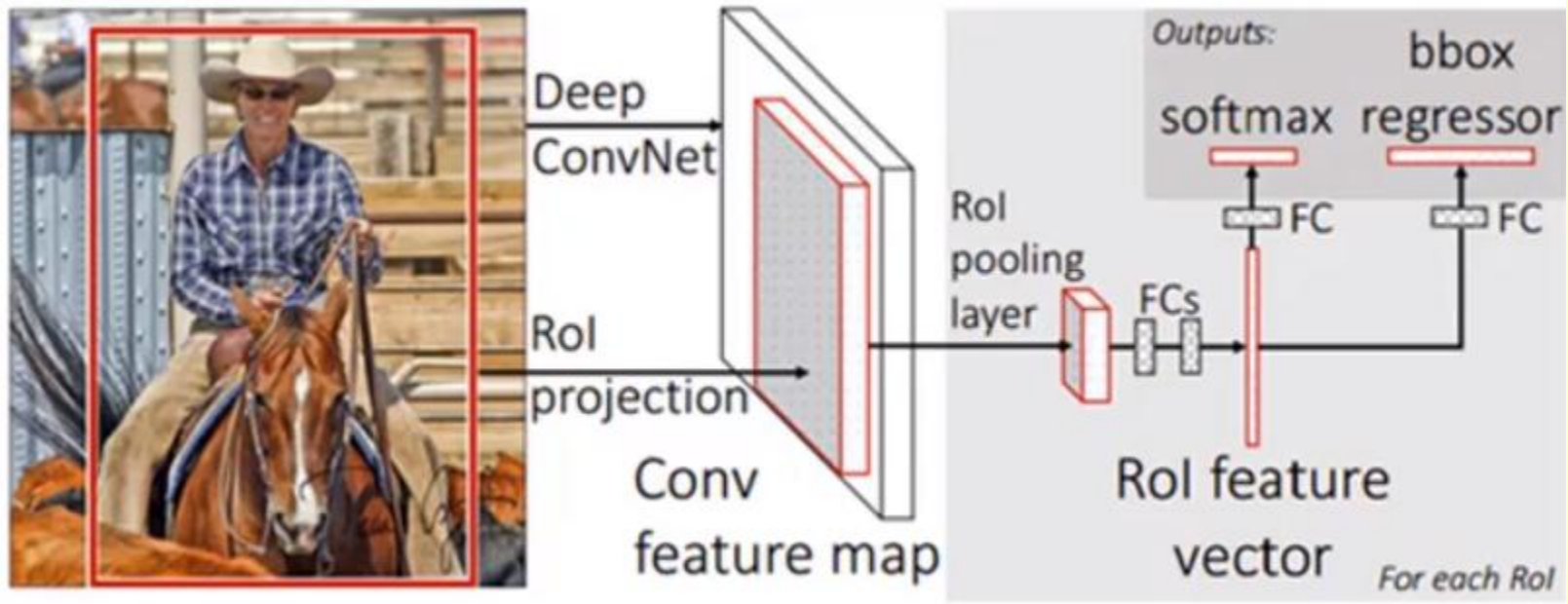
RCNN

- پیشنهاد ناحیه (region proposal)
- تغییر اندازه ناحیه برای خوراندن به CNN
- استخراج ویژگی با یک CNN
- دسته‌بندی نواحی : استفاده از SVM برای هر دسته
- انتخاب بهترین کادر پیرامون



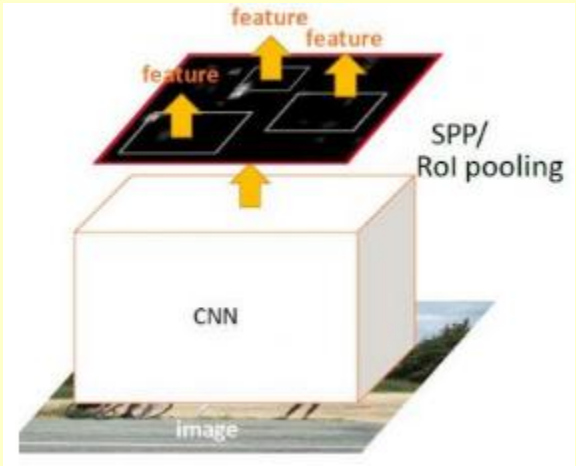
Fast RCNN

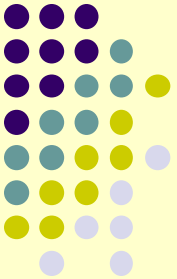
- ضعف‌های RCNN : زمان زیاد آموزش به دلیل داشتن ۳ بخش مجزای CNN، SVM و برآورد کادر پیرامون.
- تقسیم محاسبات لایه‌های CNN بین نواحی پیشنهادی متفاوت.
- جابجایی ترتیب تولید نواحی پیشنهادی و اجزای CNN .
- ابتدا اعمال CNN و سپس استخراج ویژگی برای هر ناحیه



گام‌های روش

- اعمال تصویر به CNN
- همراه با تصویر چند ناحیه مورد علاقه (RoI) وارد تصویر می‌شوند.
- استخراج یک بردار ویژگی با اندازه ثابت
- لایه تمام متصل برای هر RoI
  - خروجی Softmax برای برآورد احتمال تعلق به K دسته یا پس زمینه
  - خروجی bbox ارائه ۴ عدد برای هر k دسته شی





فصل ششم
شبکه عصبی بازگشتی (RNN)

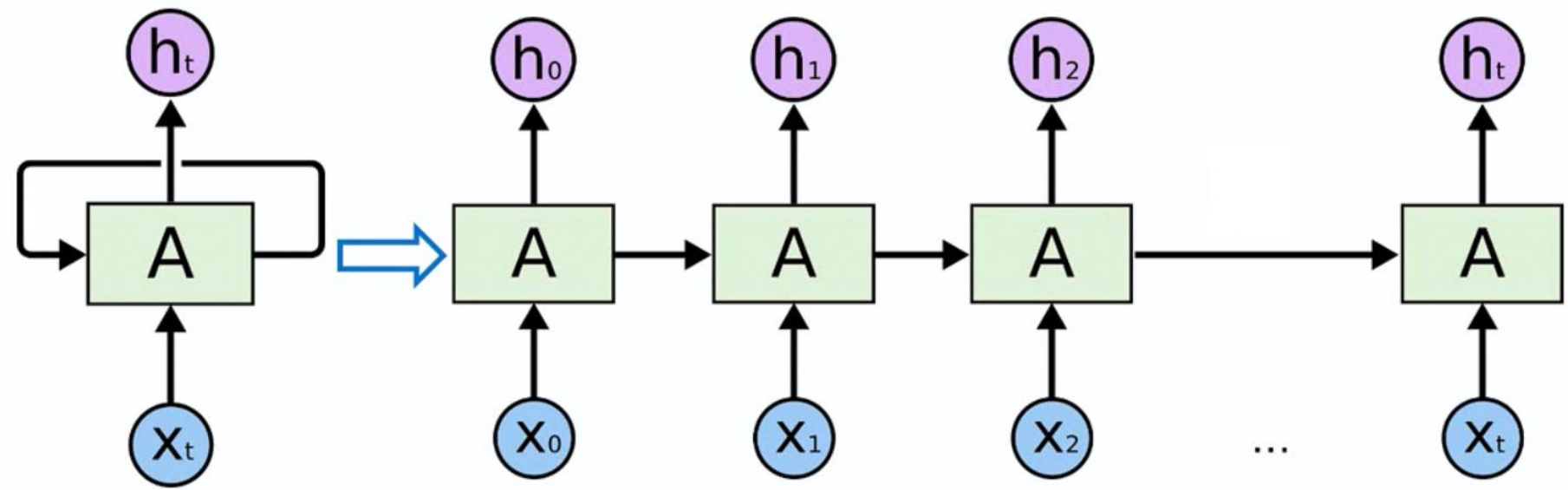
معرفی مختصر

- آموزش به کمک حلقه‌های بازگشتی : داده‌های ورودی به شبکه آموزش داده می‌شوند و اطلاعات آن برای آموزش داده‌های بعدی به شبکه باز می‌گردد.
  - استفاده از اطلاعات قبلی برای داده‌های جدید (حافظه کوتاه مدت)
  - مناسب برای داده‌های دنباله‌دار و زمان‌دار : مانند ویدئو، صوت و متن
  - وقتی کلمه یا تصویر فعلی به کلمه یا تصویر قبلی وابسته است، این شبکه کاربرد دارد.

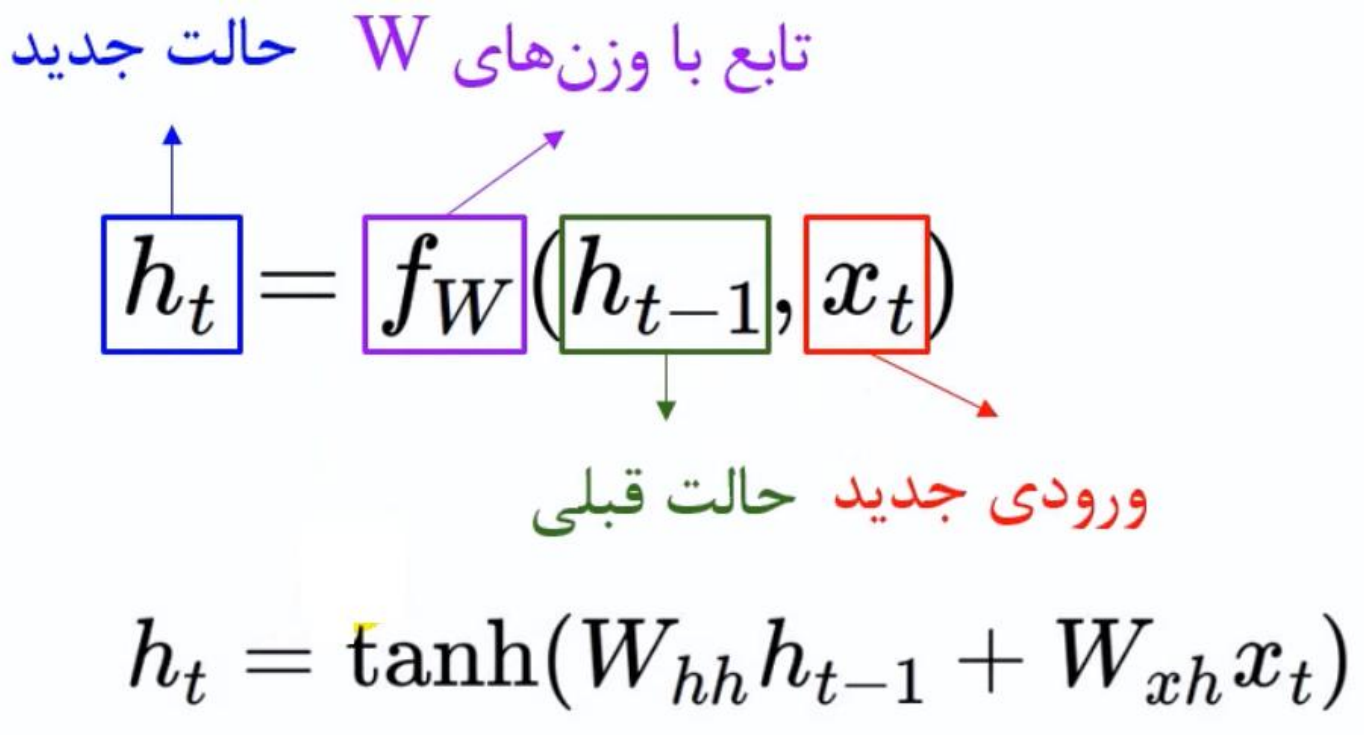
شبکه RNN باز شده

- حالت‌ها: خروجی‌های شبکه در هر زمان هستند.
- برای تولید خروجی در هر زمان ورودی‌ها و حالت‌های قبلی نقش دارند.
- ورودی و حالت تابع زمان هستند.
- فرآیند ترکیب ورودی و حالت قبل به صورت فرمول زیر است:

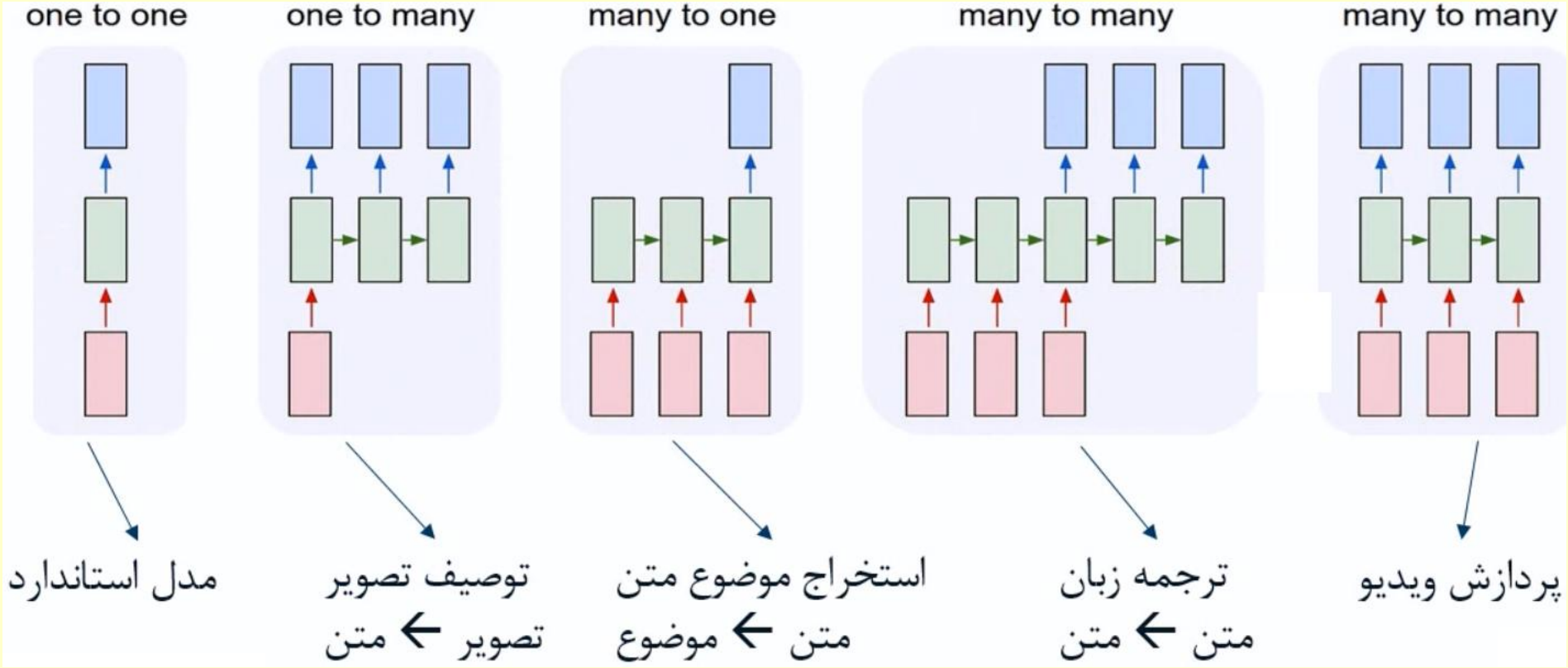
$$h_t = f_W(h_{t-1}, x_t)$$



- تحلیل رابطه:
- شبکه دو وزن دارد که یکی مربوط به ورودی و دیگری مربوط به حالت قبل است.
- تابع غیرخطی معمولا tanh است.



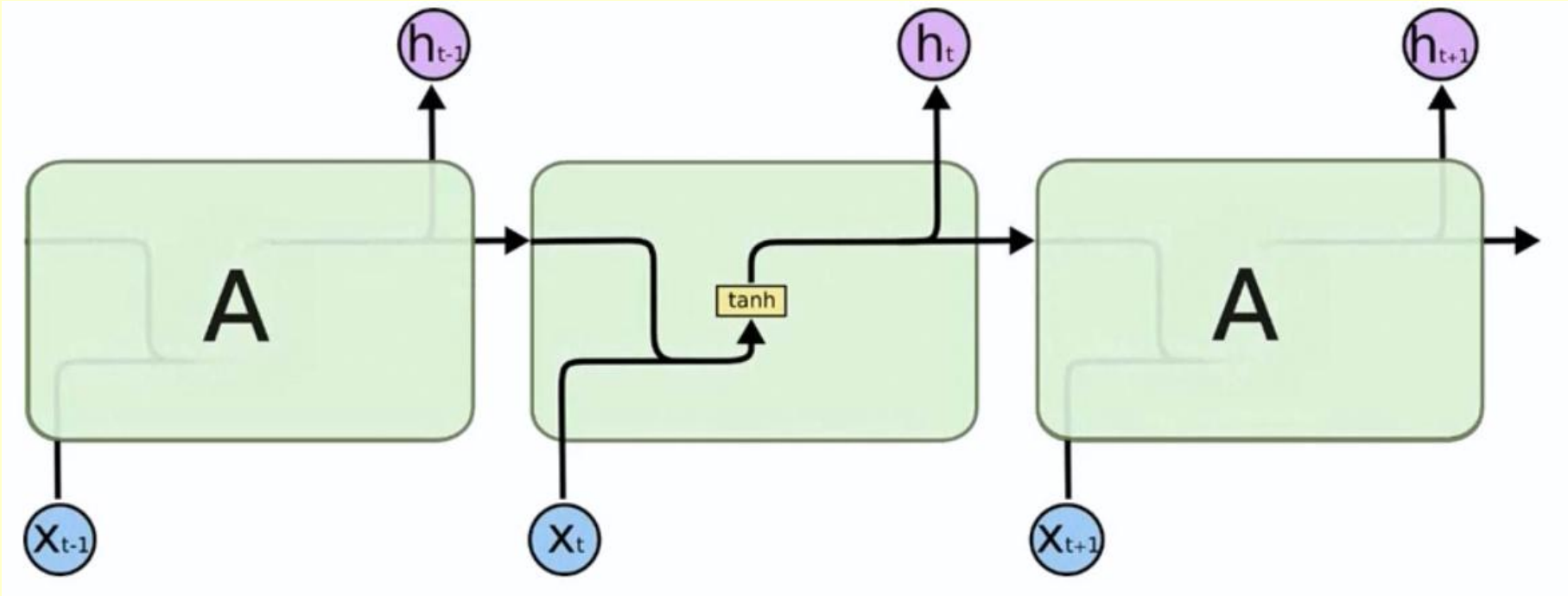
مدل‌های RNN


one to one
one to many
many to one
many to many
many to many
مدل استاندارد
توصیف تصویر
تصویر ← متن
استخراج موضوع متن
متن ← موضوع
ترجمه زبان
متن ← متن
پردازش ویدیو

مدل LSTM

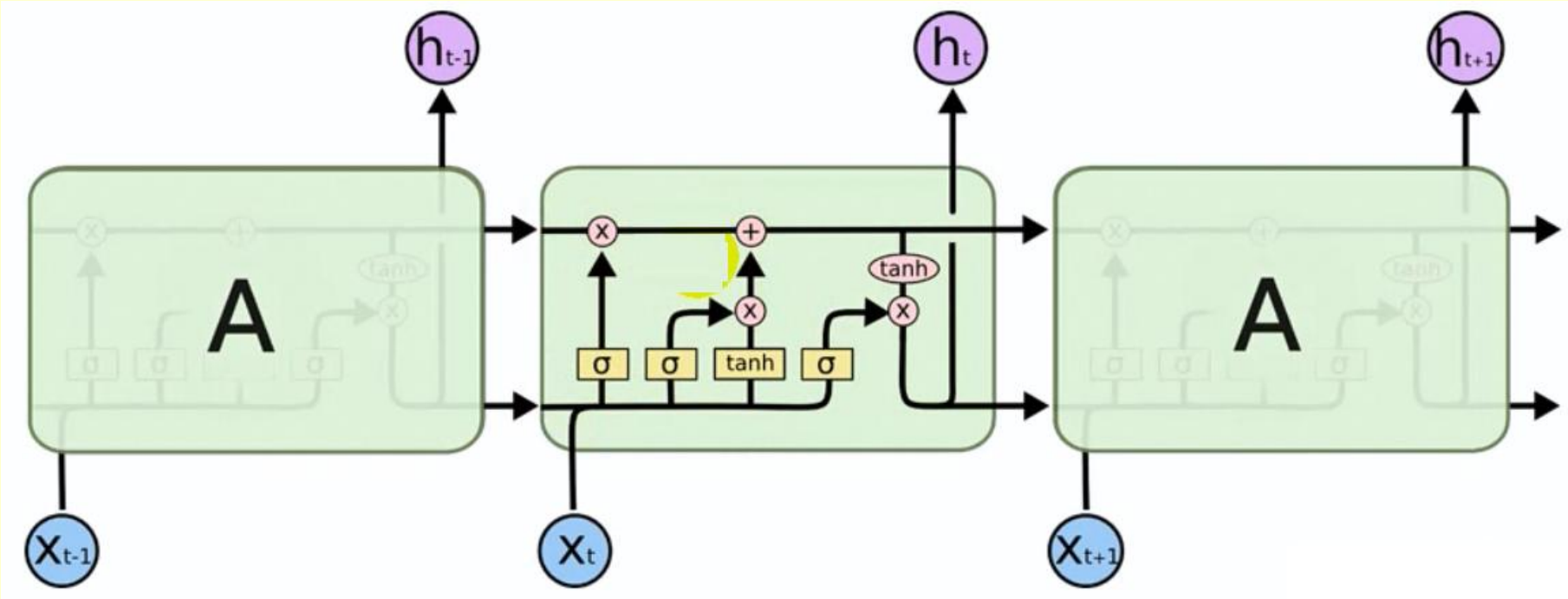
- LSTM : Long Short Term Memory : حافظه کوتاه مدت بلند
- نوع خاصی از شبکه‌های عصبی بازگشتی : بجای تابع tanh از توابع پیچیده‌تر و ترکیبی استفاده شده که موجب می‌شود حافظه کوتاه مدت مدل تنظیم شود.
- هدف LSTM : حل کردن مشکل وابستگی بلند مدت در داده‌ها : مثلا ممکن است یک کلمه به چند کلمه قبل یا چند پارگراف قبل مرتبط باشد و این ارتباط در این مدل لحاظ می‌شود.
- در بیشتر مواقع عملکرد بهتر از شبکه‌های عصبی بازگشتی استاندارد.

RNN استاندارد



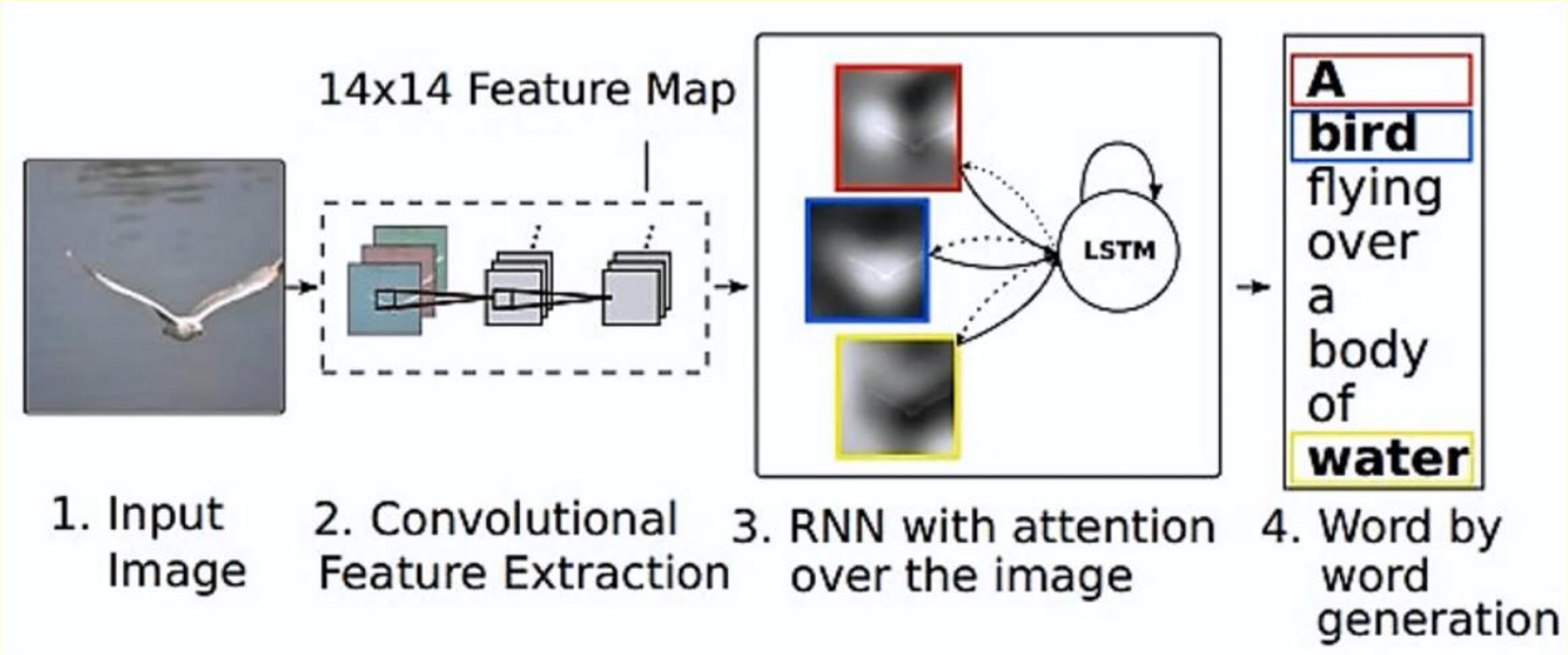
شبکه RNN با مدل LSTM

- استفاده از دروازه برای کنترل میزان تزریق ورودی یا حالت قبل در هر مسیر
- دو خروجی در هر واحد: ترکیبی از حالت‌های قبل و ورودی‌ها



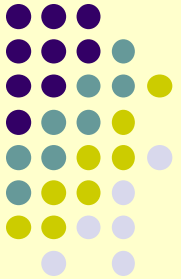
ترکیب CNN و RNN

- ایجاد توضیح متنی برای تصویر
- برای افراد نابینا



کاربردهای RNN

- تشخیص صدا و گفتار: ورودی صوت، خروجی متن صوتی یا تشخیص نوع صدا
- تشخیص موضوع و مفهوم متون: موضوع متن یا احساس موجود در متن استخراج می‌شود.
- ترجمه زبان
- درج خودکار توضیح متنی برای تصویر
- پردازش ویدئو: به کمک اطلاعات چندین فریم



فصل هفتم آموزش شبکه‌های عصبی

مراحل آموزش

۱. آماده‌سازی داده‌های آموزش
۲. انتخاب معماری مناسب شبکه
۳. الگوریتم آموزش و بهینه‌سازی
۴. روش‌های بهبود آموزش

اصول آماده‌سازی داده‌ها

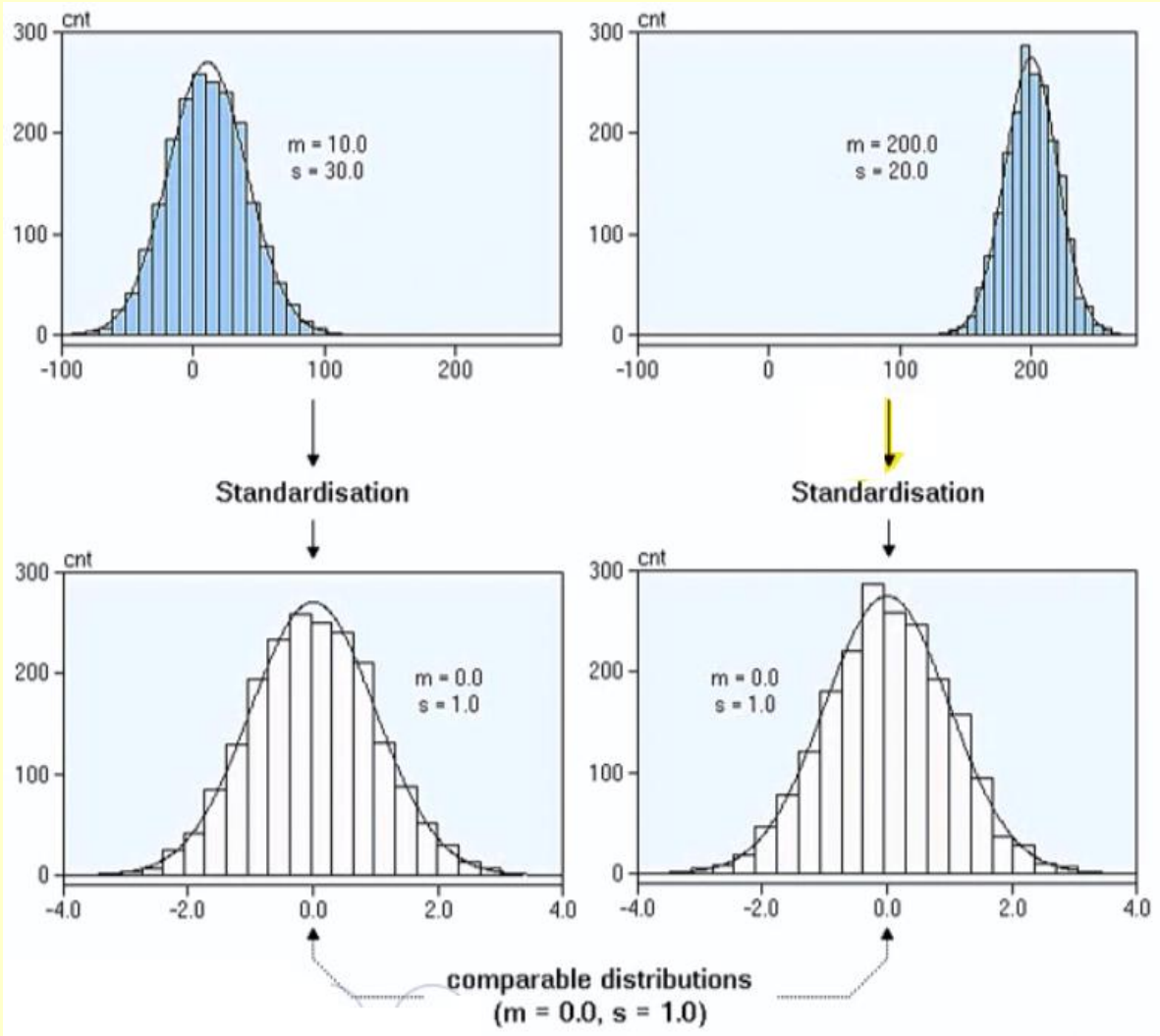
- داده‌ها باید برای آموزش شبکه مناسب باشند: به این معنی که دارای اطلاعاتی نباشند که شبکه را گمراه کند.
- داده بیشتر= آموزش بهتر
- حذف داده‌های ناقص و مخدوش : این داده‌ها موجب گمراهی شبکه می‌شوند.
- پیش‌پردازش داده‌ها (یکسان‌سازی شرایط) : مثلا نرمال‌سازی بردارهای ورودی (یا ویژگی‌ها) بین صفر تا یک. این نرمال‌سازی موجب می‌شود که مقادیر بزرگ‌تر همانند مقادیر کوچک‌تر در شبکه تاثیر گذار باشند و یک ویژگی خاص ارزش بیشتری در آموزش نداشته باشد.
- استفاده از روش‌های افزایش داده (Data augmentation)

پیش‌پردازش داده‌ها

- هدف
  - حذف اطلاعات و تغییرات نامطلوب و مشابه‌سازی داده‌ها
- مثال
  - حذف تغییرات نور و یکسان‌سازی محدوده شدت روشنایی تصاویر
  - یکسان‌سازی مقیاس در داده‌های عددی
  - یکی از راه‌های یکسان‌سازی، استاندارد کردن داده‌ها است.
  - تغییر داده‌ها برای داشتن میانگین صفر و واریانس ۱ با استفاده از رابطه زیر

$$x_{standardized} = \frac{(x - \text{mean}(x))}{\text{std}(x)}$$

استاندارد سازی داده‌ها



نرمالیزه کرده داده‌ها

- انتقال محدوده داده‌ها بین صفر تا یک

$$x_{normalized} = \frac{(x - \min(x))}{(\max(x) - \min(x))}$$

افزایش داده‌ها

- ایجاد تغییرات بر روی داده‌های فعلی و تولید داده‌های جدید
- مثال: جابجایی، تغییر اندازه، چرخش، تغییر شکل و رنگ

انتخاب معماری شبکه

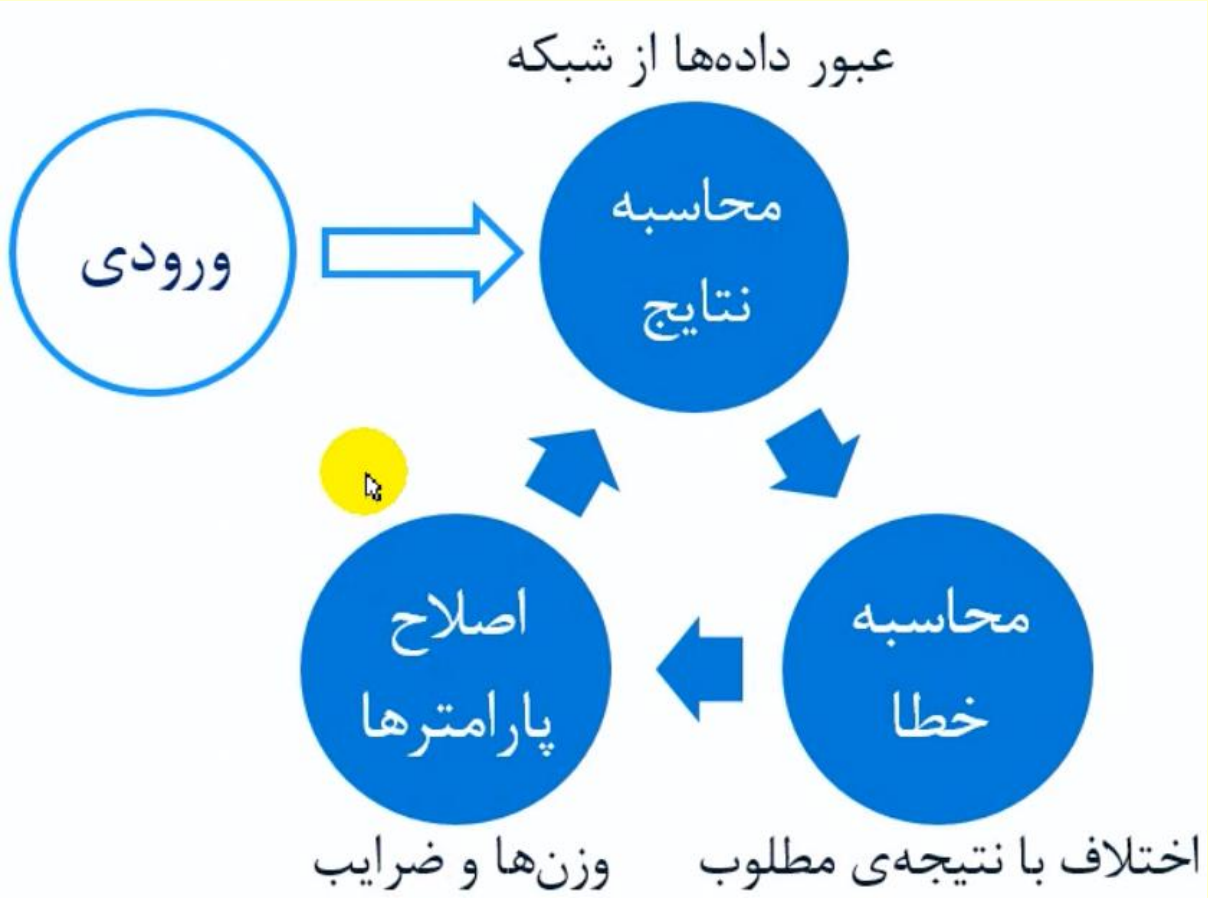
- نوع داده‌ها موجب انتخاب معماری بازگشتی، تمام متصل یا کانولوشنی می‌شود.
- تعداد پارامترها در طراحی مهم است. مثلا چند فیلتر استفاده کنیم یا تعداد نورون‌های هر لایه چقدر باشد.
- نحوه چینش لایه‌ها چگونه باشد.
- تئوری قوی در این بخش نداریم. اما اصول کلی داریم که می‌توان از آن‌ها کمک گرفت.
- تعداد لایه‌ها و نورون‌ها (فیلترها)
  - شروع با تعداد کم
  - افزایش تا جایی که نتیجه بهبود نیابد.

انتخاب نوع شبکه

- داده‌های یک بعدی (سیگنال، بردار) : شبکه FC یا AE
- داده‌های چند بعدی (تصویر، تنسور): شبکه CNN
- داده‌های وابسته به زمان (صوت، ویدئو، متن): شبکه RNN
- تنظیم پارامترها، چینش لایه‌ها و تعداد نورون‌ها با آزمون و خطا بدست می‌آیند.
- هنر و خلاقیت معمار شبکه حائز اهمیت است. مثلا در تصویر می‌توان از شبکه RNN یا ترکیب CNN و RNN بهره برد.

آموزش شبکه

- فرآیند چرخشی و گام به گام
- تکرار چرخه آموزش تا رسیدن به نتایج مطلوب



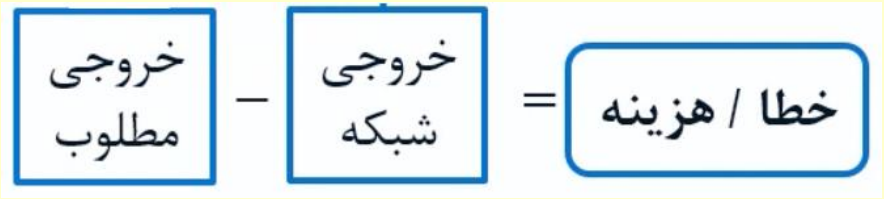
محاسبه خطا

- عبور داده‌ها از شبکه و مقایسه با نتیجه مطلوب

- خروجی نهایی :

$$\tilde{y} = f(W_2(\overbrace{W_1(\underbrace{W_0 * X}_{H_1})}^{H_2}))$$

- خروجی نهایی شبکه باید با خروجی مطلوب مقایسه شود:



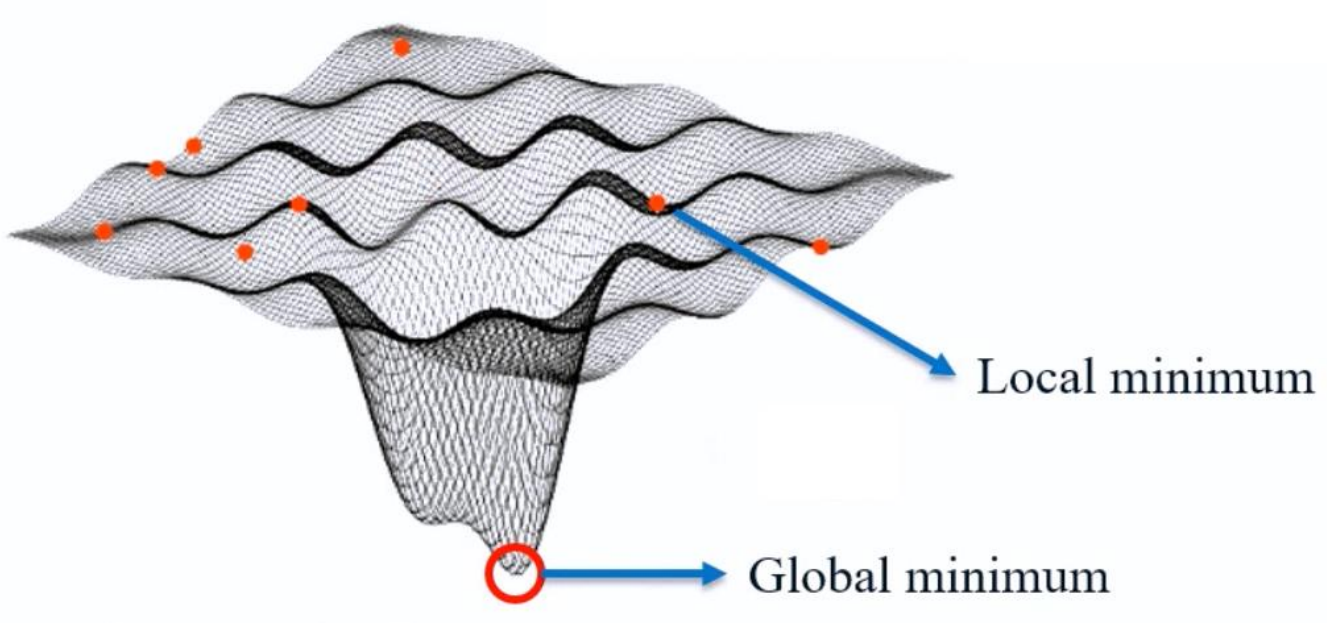
محاسبه خطا

- نحوه محاسبه خطا با روش‌های مختلفی انجام می‌شود.
- انتخاب نحوه محاسبه خطا معادل انتخاب تابع هزینه است.
- نمونه‌هایی از تابع هزینه (Cost/ Loss Function)
  - Mean Squared
  - Cross- entropy
  - Hinge
  - Softmax

$$\text{Mean squared: } \mathrm{L}(y, \tilde{y}) = \frac{1}{N}\sum_{i}(y_i - \tilde{y}_i)^2$$

بهینه‌سازی

- پس از محاسبه خطا نیاز به اصلاح پارامترهای شبکه است.
- بهینه‌سازی: حرکت قدم به قدم به سمت کم‌ترین مقدار خطا است. یعنی اصلاح و بهینه‌سازی وزن‌ها برای رسیدن به حداقل خطا است.
- مقادیر مختلف تابع هزینه (مجموعه تمام مقادیر خطاهای ممکن) یک سطح ناهموار را تشکیل می‌دهند. بعد سطح ناهموار بستگی به بعد خطا دارد.
- هدف: رسیدن به عمیق‌ترین دره است.



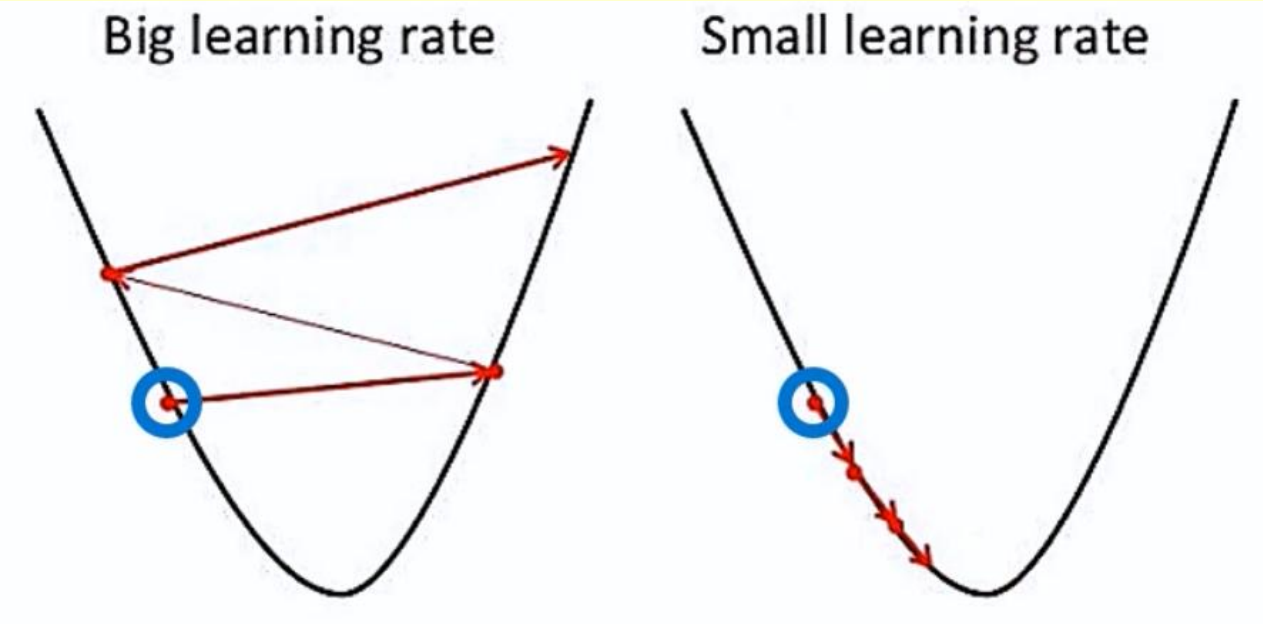
بهینه‌سازی

- با استفاده از گرادیان می‌توان به سمت Global Minimum حرکت کرد. البته ممکن است در Local Minimum گیر کنیم.
- الگوریتم‌هایی برای رهایی از Local Minimum ارائه شده‌اند.
- SGD
- SGD+ Momentum
- RMSprop
- Adagrad
- Adadelta
- Adam

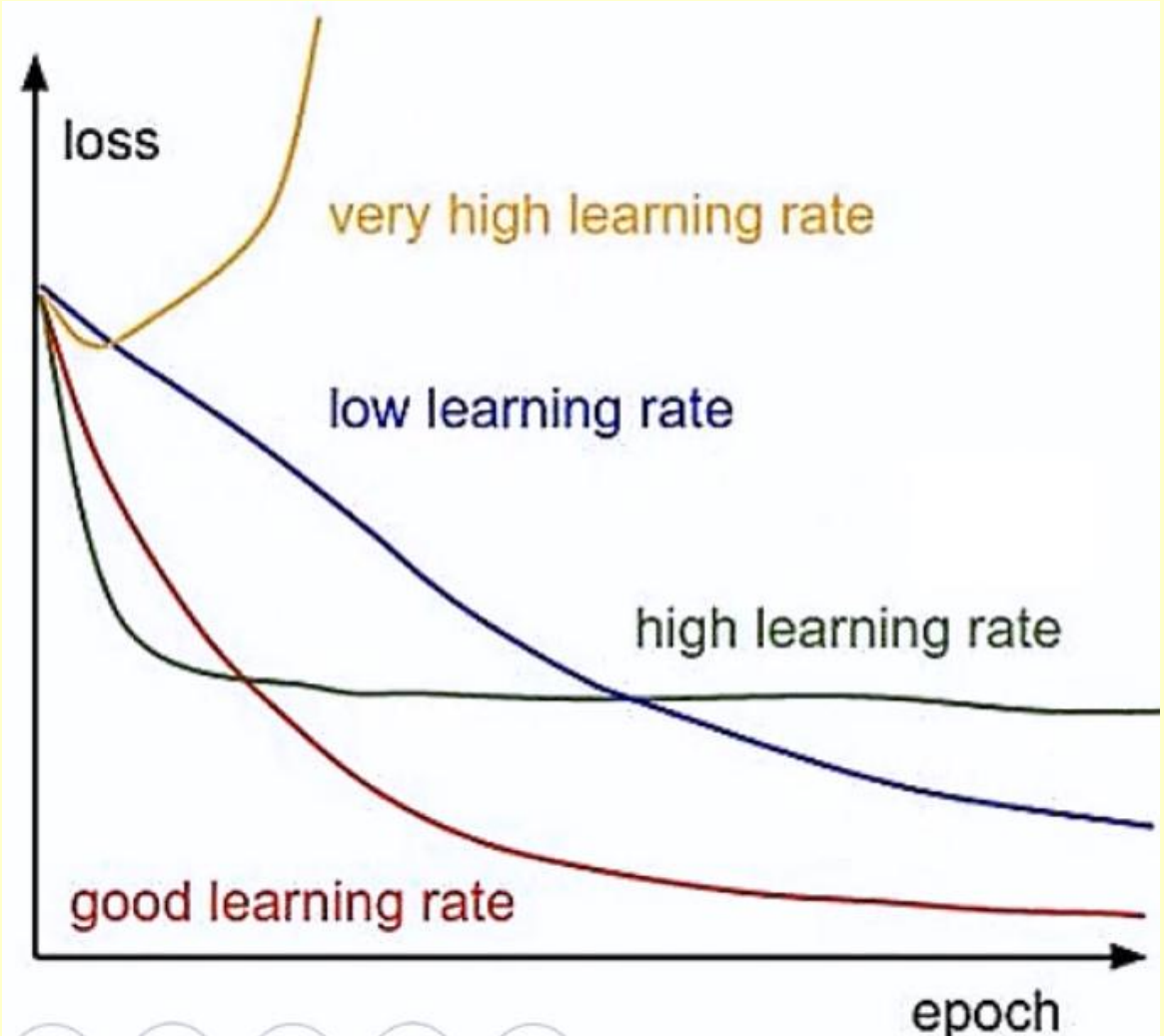
http://ruder.io/optimizing-gradient-descent/

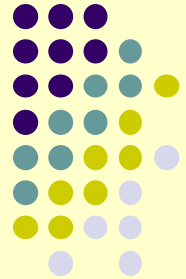
نرخ آموزش

- تعریف: اندازه گام‌ها برای رسیدن به حداقل خطا
- اندازه گام نباید خیلی بزرگ یا خیلی کوچک باشد.



اثر مقادیر مختلف نرخ آموزش





روش‌های بهبود آموزش

- گیر کردن در Local Minimum
- عدم اصلاح مناسب وزن‌ها
- سرعت کم در آموزش
- عدم دقت مناسب در نتایج نهایی
- روش‌های بهبود آموزش
  - Batch Normalization
  - Dropout
  - Transfer Learning