دانشگاه شهید چمران اهواز

کلاس برنامه نویسی پایتون

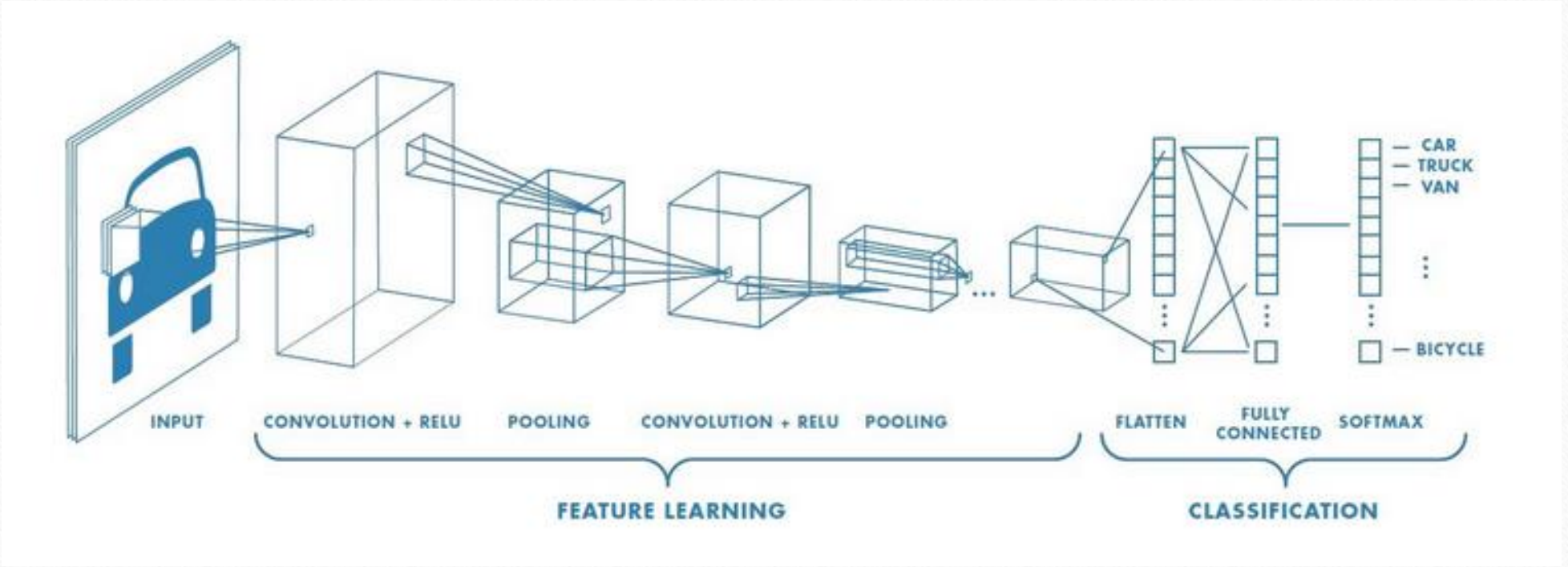
موضوع:شبکه عصبی کانولوشن(CNN)

ارائه دهنده: مرجان مودت

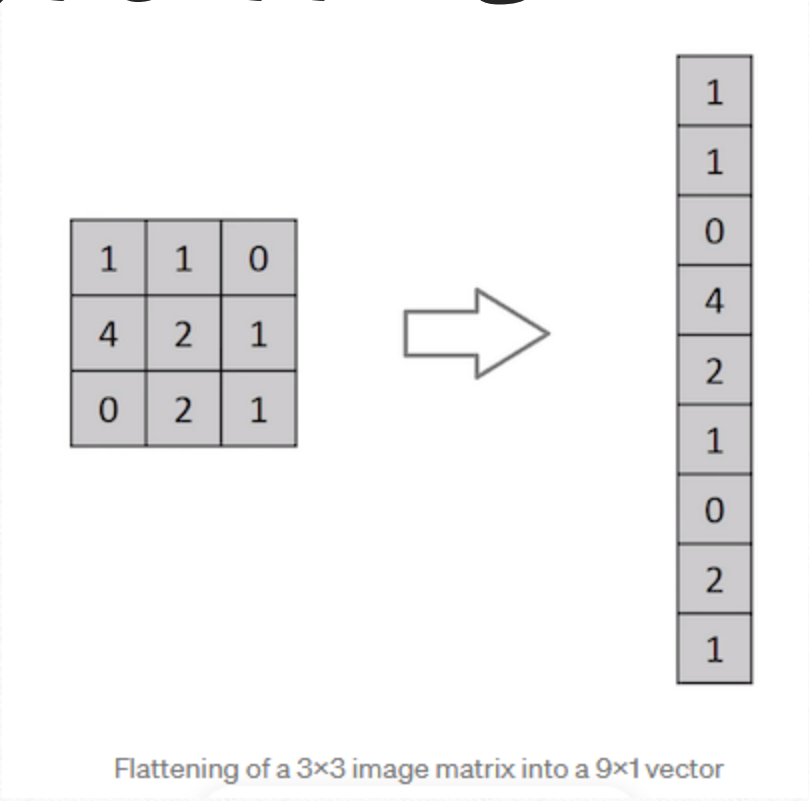
انواع شبکه عصبی

- MLP
- **(Convolutional Neural Network) CNN**
- RNN
- GAN
- LSTM
- و...

شبکه عصبی کانولوشن چیست؟



تفاوت شبکه عصبی کانولوشن و پرسپترون



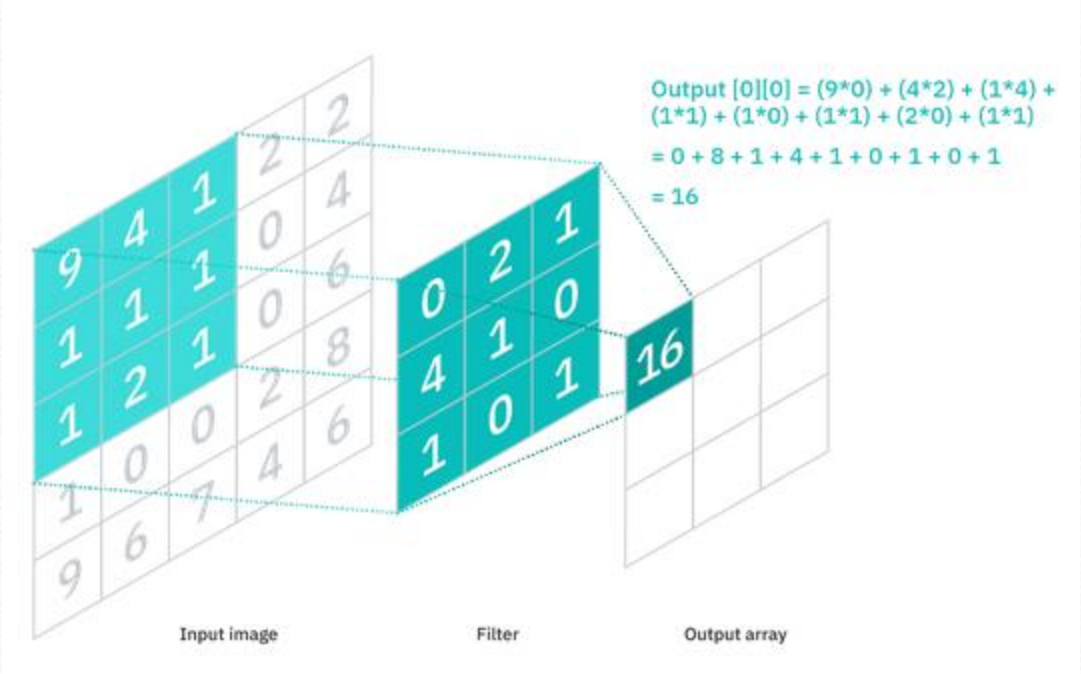
Flattening of a 3×3 image matrix into a 9×1 vector

اجزای شبکه عصبی کانولوشن

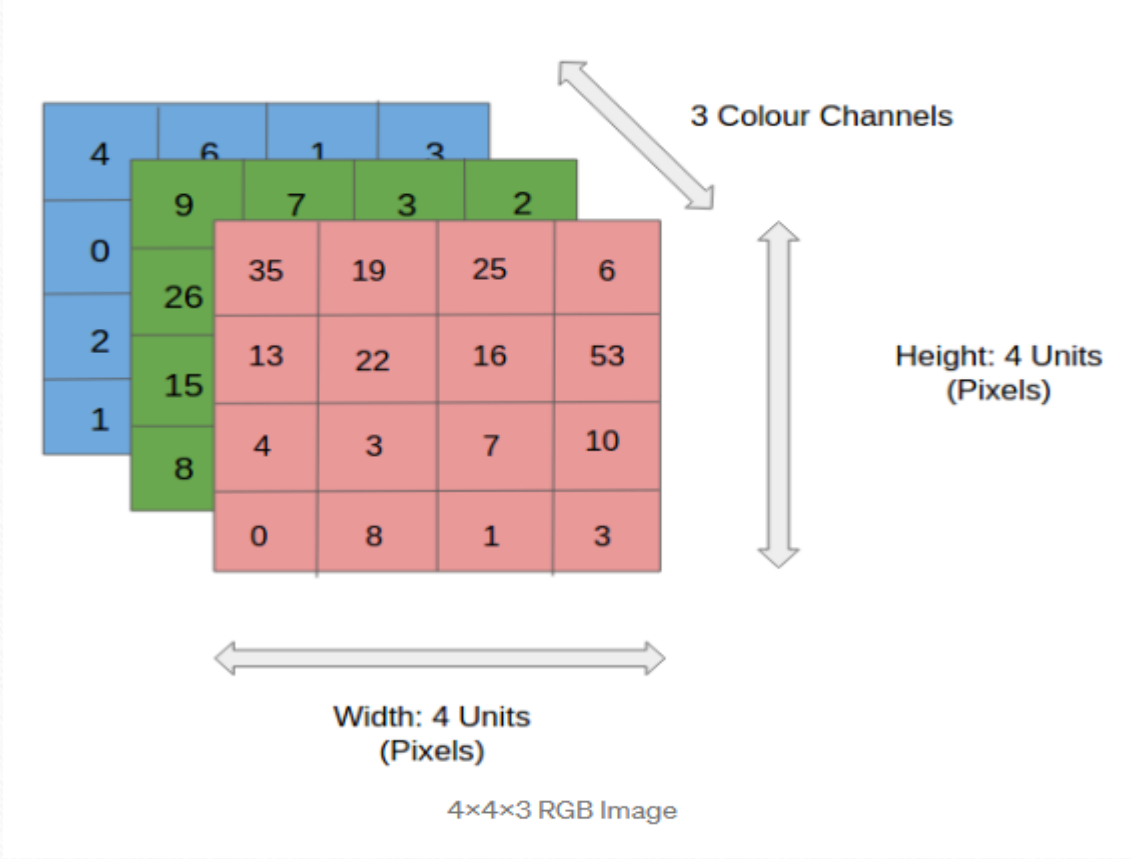
سه نوع لایه اصلی دارد:

1) لایه کانولوشن
2) لایه ادغام
3) لایه کاملا متصل

۱) لایه کانولوشن

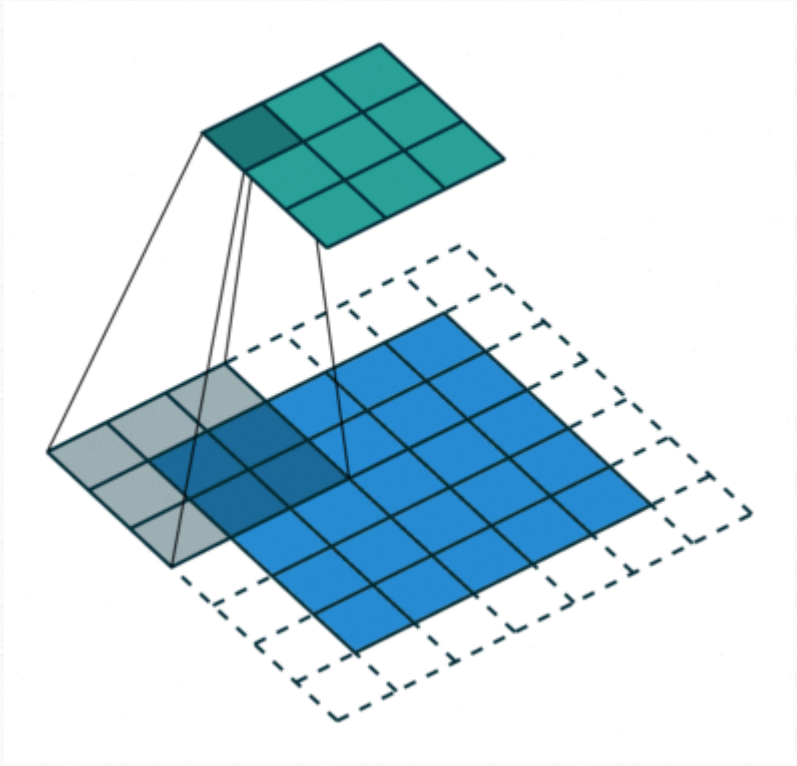


کانال رنگی


3 Colour Channels
4
6
1
3
9
7
3
2
0
35
19
25
6
26
2
13
22
16
53
15
1
4
3
7
10
8
0
8
1
3
Height: 4 Units
(Pixels)
Width: 4 Units
(Pixels)
4×4×3 RGB Image

سه هایپر پارامتر موثر بر خروجی

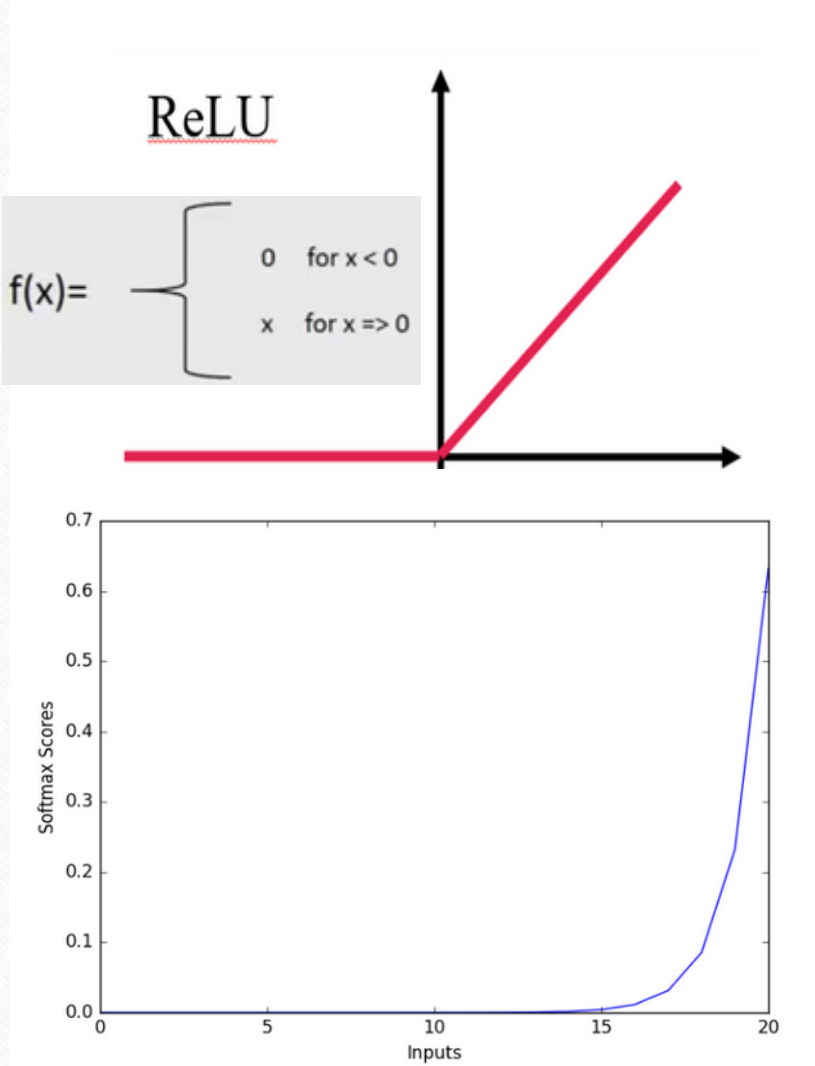
1. تعداد فیلترها
2. گام فاصله (شکل ۱)
3. Zero padding
    1. Valid
    2. Same
    3. full



شکل ۱: گام ۲

تابع فعال ساز:

1) Relu
2) Softmax(دسته بندی)



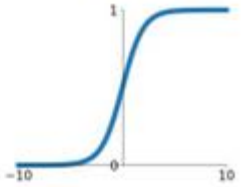
Softmax:

این تابع جهت اسکیل خروجی به بازه صفر و یک استفاده می کنیم که از این طریق چگالی پیش بینی برای همه کلاس ها مدل مشخص میشه و مجموع این چگالی برای همه کلاس یک است.

Activation Functions

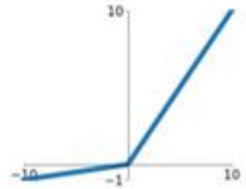
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



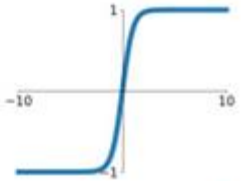
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

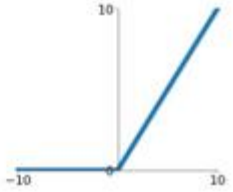


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

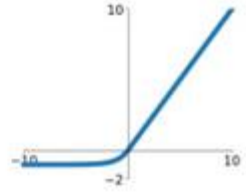
ReLU

$$\max(0, x)$$



ELU

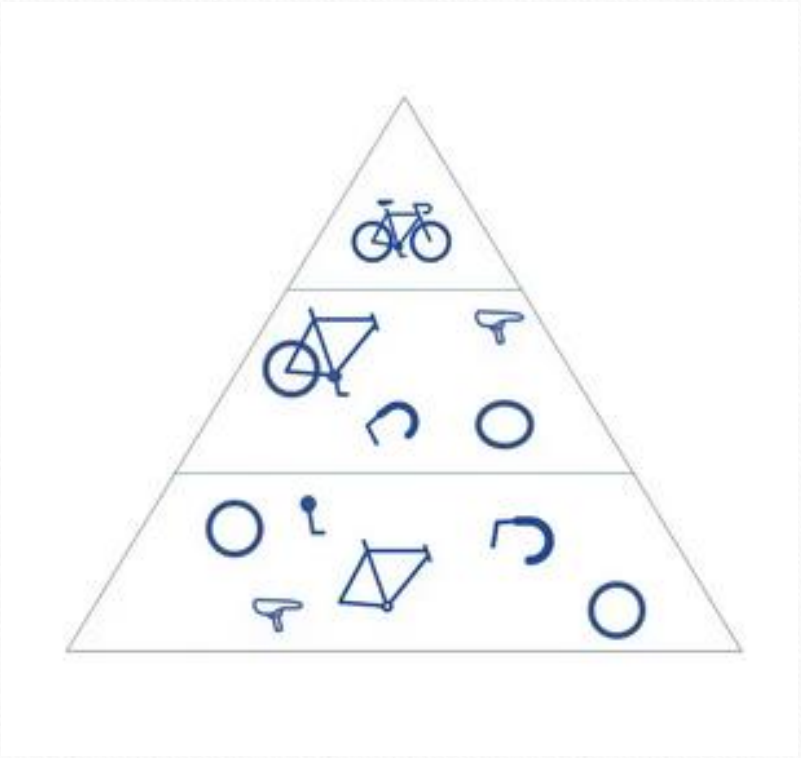
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



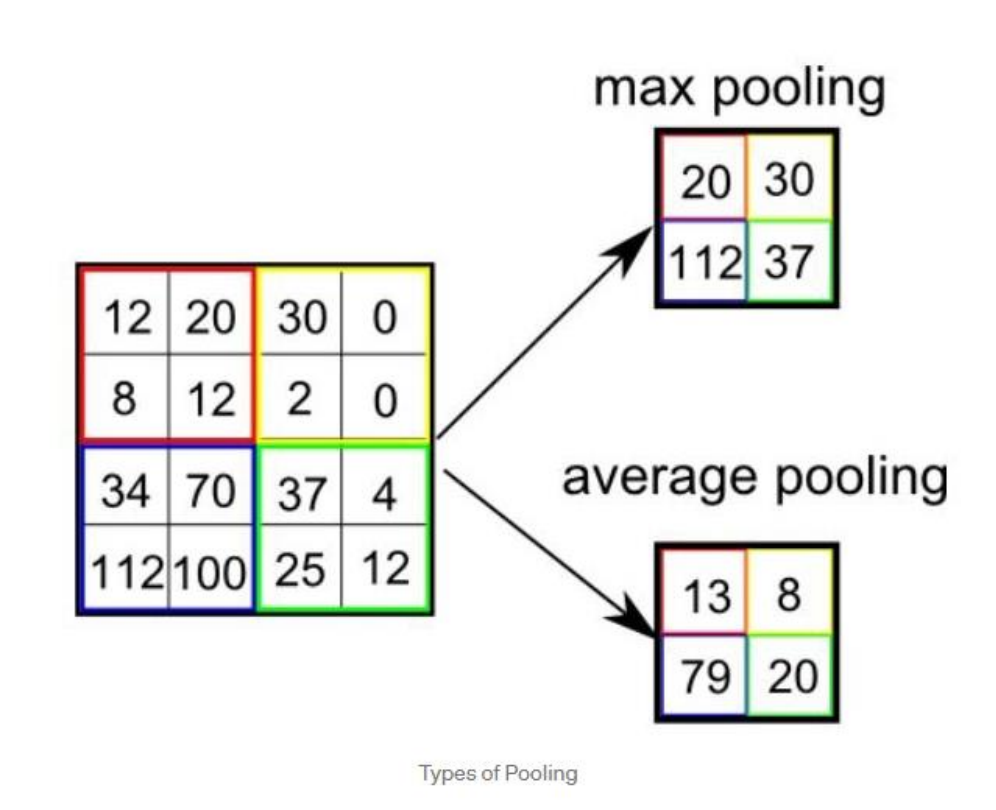
Sigmoid در لایه خروجی در حین انجام پیش بینی های باینری استفاده می شود.

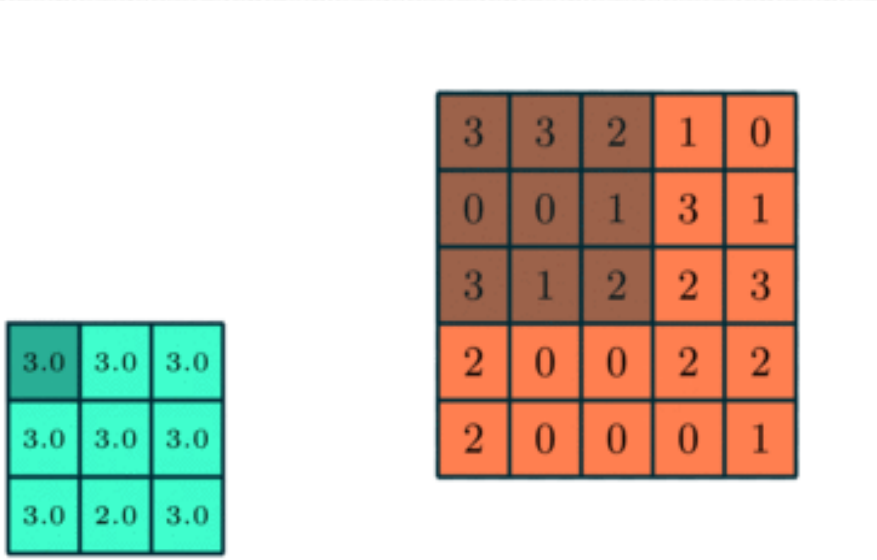
Softmax در لایه خروجی در حین انجام پیش بینی های چند کلاسه استفاده می شود.

ساختار سلسله مراتبی cnn

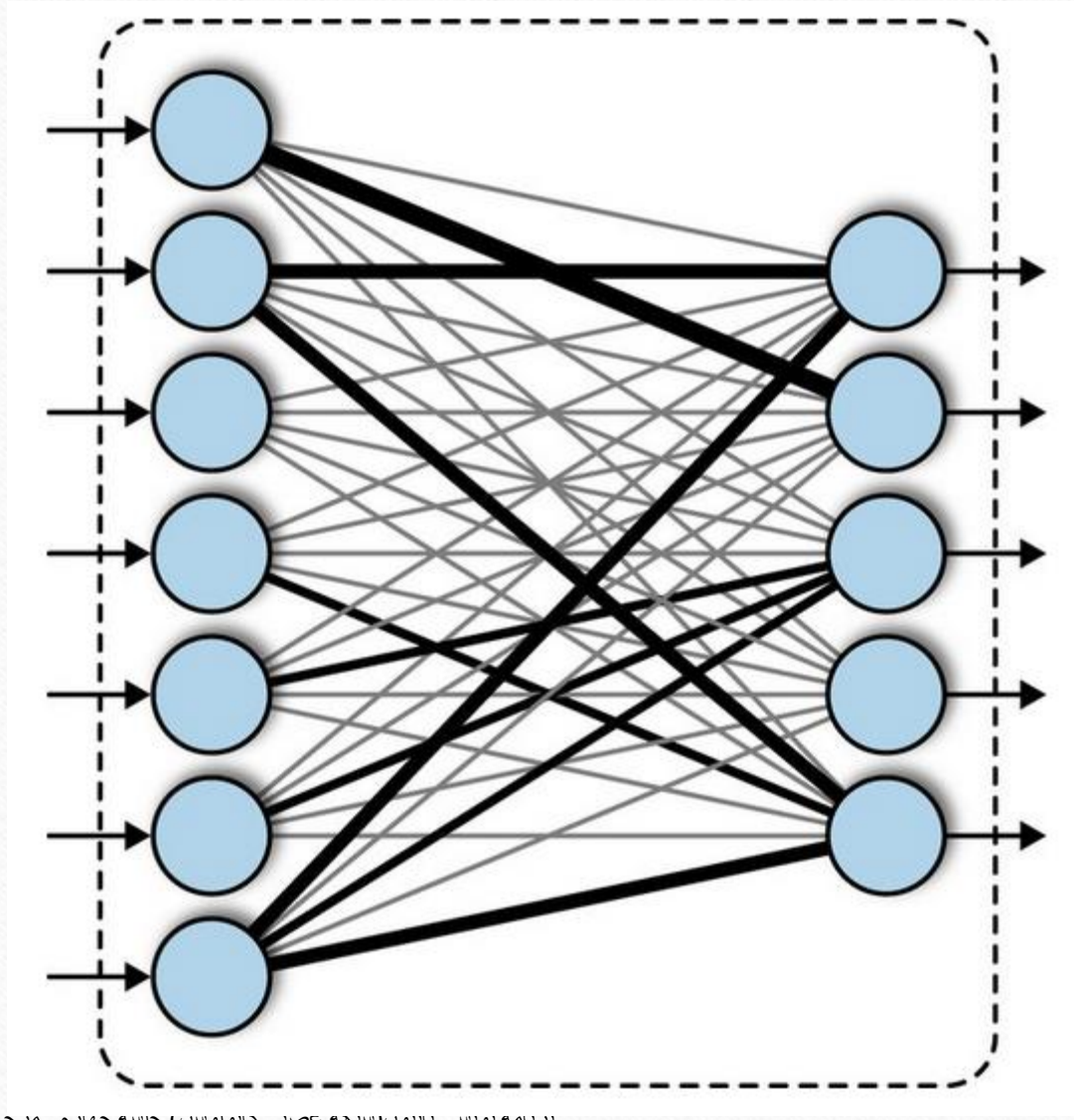


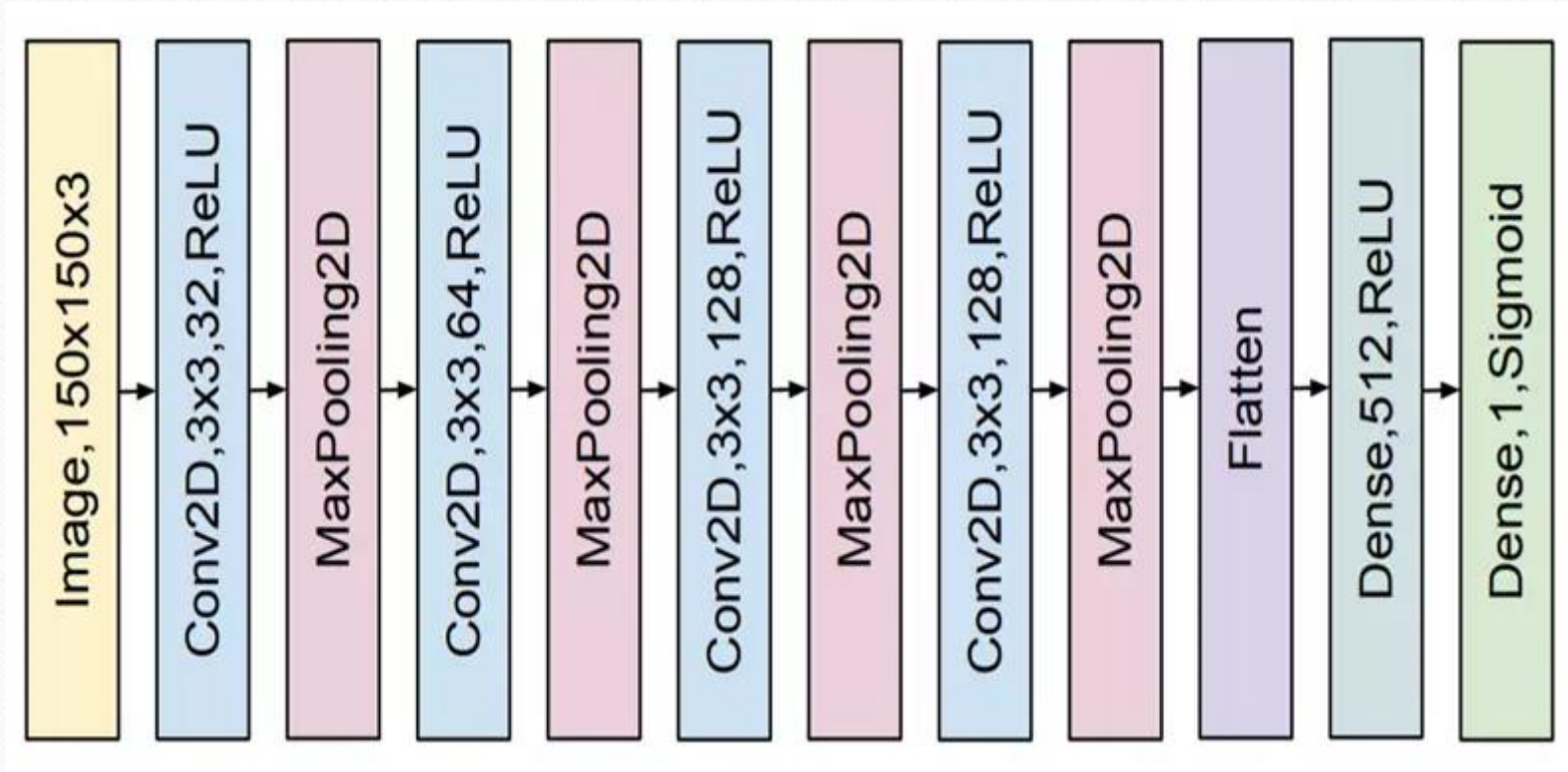
۲) لایه ادغام


max pooling
20 30
112 37
12 20 30 0
8 12 2 0
34 70 37 4
112 100 25 12
average pooling
13 8
79 20
Types of Pooling


3.0 3.0 3.0
3.0 3.0 3.0
3.0 2.0 3.0
3 3 2 1 0
0 0 1 3 1
3 1 2 2 3
2 0 0 2 2
2 0 0 0 1

۳) لایه کاملا متصل




Image,150x150x3
Conv2D,3x3,32,ReLU
MaxPooling2D
Conv2D,3x3,64,ReLU
MaxPooling2D
Conv2D,3x3,128,ReLU
MaxPooling2D
Conv2D,3x3,128,ReLU
MaxPooling2D
Flatten
Dense,512,ReLU
Dense,1,Sigmoid

تفاوت پارامتر و هایپرپارامتر

- پارامتر
  - مقداردهی اولیه
  - به روزرسانی
- هایپرپارامتر
  - باید اضافه شود
  - تنظیم

ادامه هاییپرپارامترها

- تعداد لایه های پنهان
- Dropout
- نرخ یادگیری
- Momentum
- Epochs(تعداد دورها)
- Batch size (اندازه دسته ها)

- تعداد لایه های پنهان
  - زیاد کردن لایه ها تا کمتر نشدن خطا تست

- Dropout
  - از ۰.۲ تا ۰.۵
  - زیاد شدن باعث کم شدن قدرت یادگیری
  - جلوگیری از بیش برازش(قدرت تعمیم)

- نرخ یادگیری
  - کم: کند و همگرا
  - زیاد: سریع و نوسانی

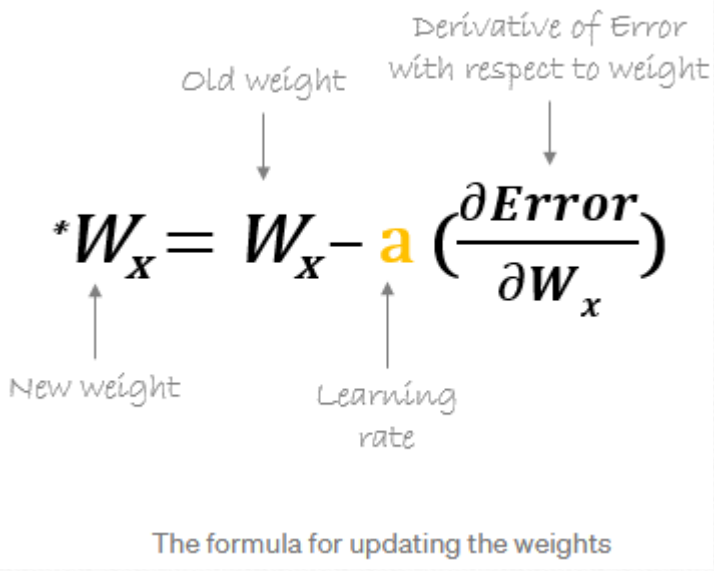
- Momentum
  - جلوگیری از نوسانات
  - ۰.۵ تا ۰.۹

- Epochs(تعداد دورها)
  - تعداد دفعاتی کل داده آموزشی به شبکه آموزش داده می شود.
  - افزایش تا جایی که دقت اعتبارسنجی شروع به کاهش کند.

- Batch size (اندازه دسته ها)
  - داده ها به صورت دسته های چندتایی به شبکه داده شود.
  - پیش فرض ۳۲ اما ۶۴ و ۱۲۸و ۲۵۶ و... امتحان کنید.

تابع بهینه ساز

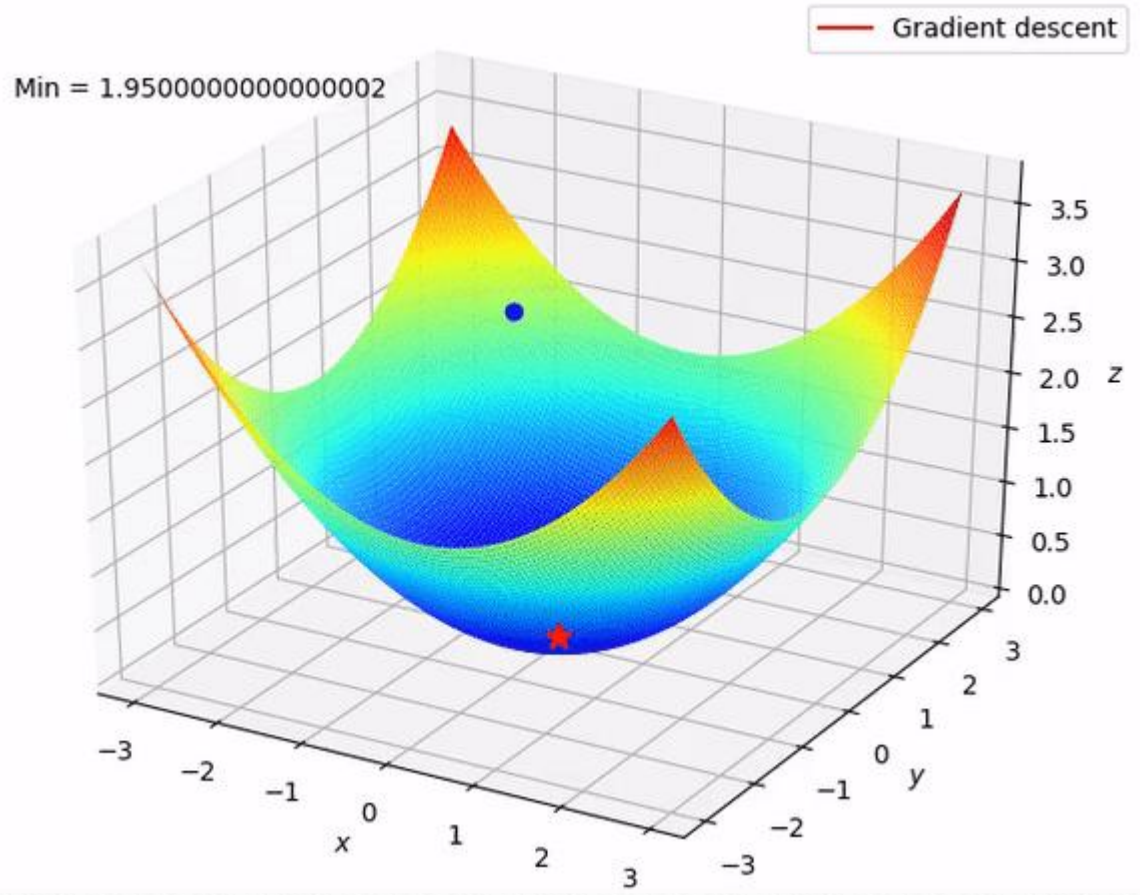
- تغییر وزن و نرخ یادگیری
- کاهش تلفات

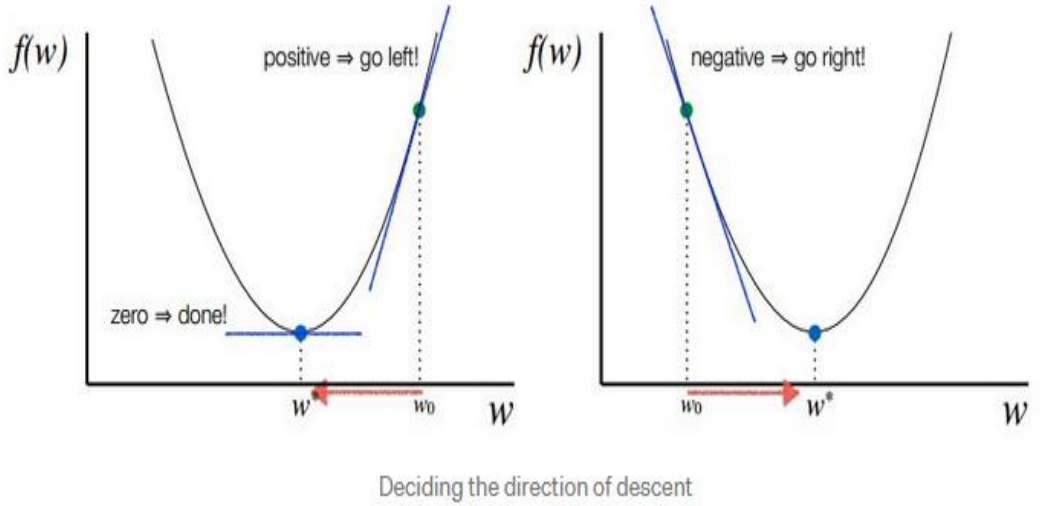


The formula for updating the weights

Gradient Descent (1

یافتن حداقل های جهانی برای یک مسئله بهینه سازی محدب با استفاده از گرادیان نزولی

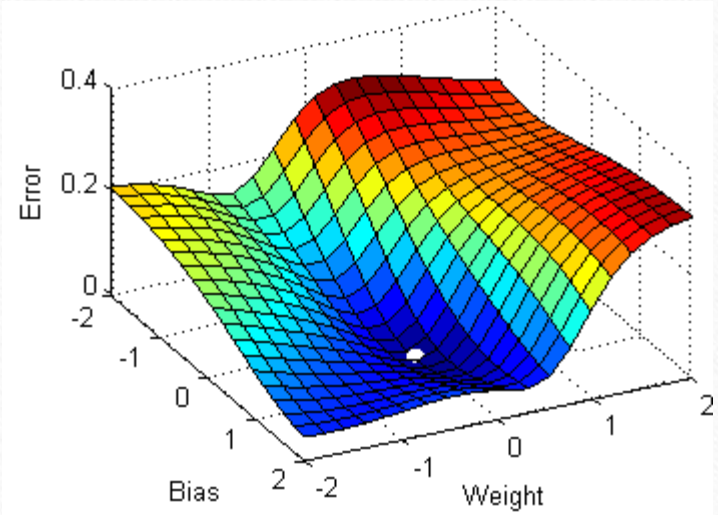




Deciding the direction of descent

Stochastic Gradient Descent(2

همگرایی در حداقل های جهانی با استفاده از SGD برای داده های غیر محدب



3) Adagrad

Adagrad برای برخورد با داده های پراکنده مناسب است.

4) Adadelta

از کاهش بی‌نهایت نرخ یادگیری مراقبت می‌کند.

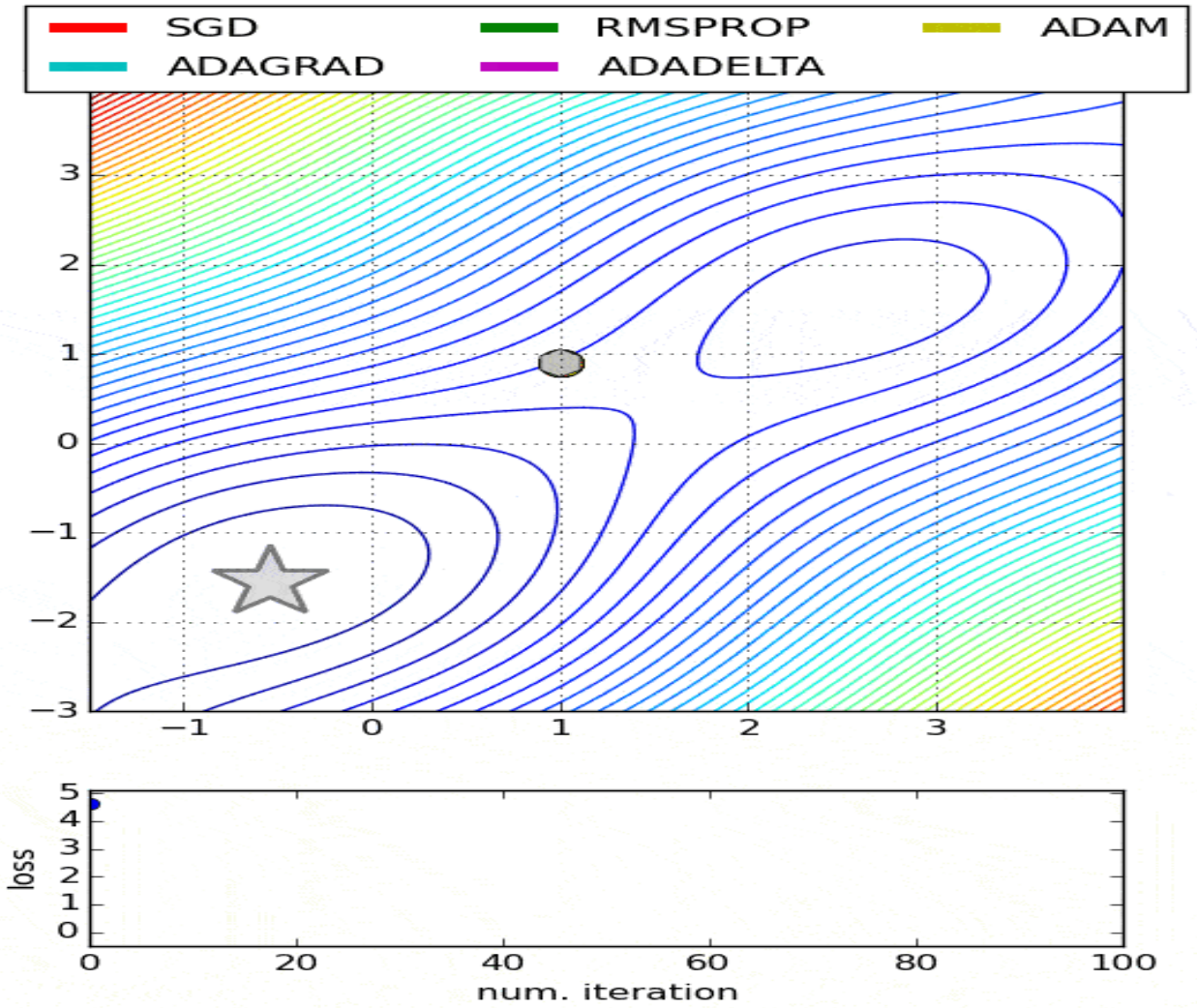
5) RMSprop

شبیه به آدادلتا

6) *Adam*

از ترکیب Gradient Descent با Momentum و RMSprop استفاده می کند ولی محاسبات پر هزینه است.

تابع بهینه ساز



تابع هزینه(loss)

- mean_square_error
  - رگرسیون
- categorical_crossentropy
  - طبقه بندی
- Binary Cross-Entropy
  - طبقه بندی

برنامه نویسی

1. پیش پردازش دیتاست(ارقام دست نویس)
2. ساخت مدل
3. تنظیم هایپرپارامترها و مقداردهی پارامترها
4. آموزش مدل
5. تست مدل (دقت و خطا)

سپاس از توجه شما دوستان