

قسمت اول: مقدمه ای بر شبکه‌های عصبی و شبکه‌های عصبی کانولوشنالی در

Keras کتابخانه

Alireza Akhavanpour

Akhavanpour.ir
CLASS.VISION

Thursday, February 7, 2019

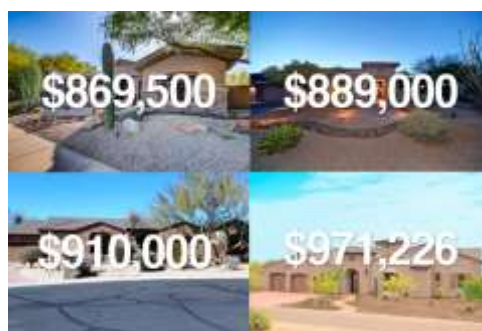
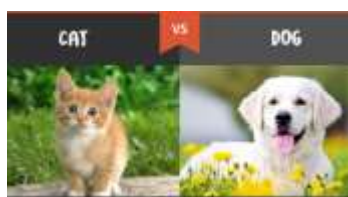
Keras مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی
علیرضا اخوان پور



چشم انداز دوره:



6



Keras



python™

Keras مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی
علیرضا اخوان پور



❑ مقدمه:

- مقدمه ۱: مقدمه‌ای بر یادگیری ماشین
- مقدمه ۲: تصویر به عنوان داده
- مقدمه ۳: طبقه بندی تصویر در پایتون

❑ شبکه‌های عصبی:

- بخش ۱: مقدمه‌ای بر پرسپترون
- بخش ۲: مقدمه‌ای بر شبکه‌های عصبی
- بخش ۳: تابع هزینه
- بخش ۴: پس انتشار خطا و گرادیان کاهشی

❑ پیاده سازی در Keras:

- بخش ۵: پیاده سازی طبقه بندی اعداد فارسی در کراس
- بخش ۶: شبکه های عصبی کانولوشنالی



سازمان اسناد و کتابخانه ملی
جمهوری اسلامی ایران

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

مقدمه ۱:

مقدمه‌ای بر یادگیری ماشین

Introduction to Machine Learning



سازمان اسناد و کتابخانه ملی
جمهوری اسلامی ایران

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

Image Classification: a core task in Computer Vision



(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

→ **cat**

Challenges: Illumination



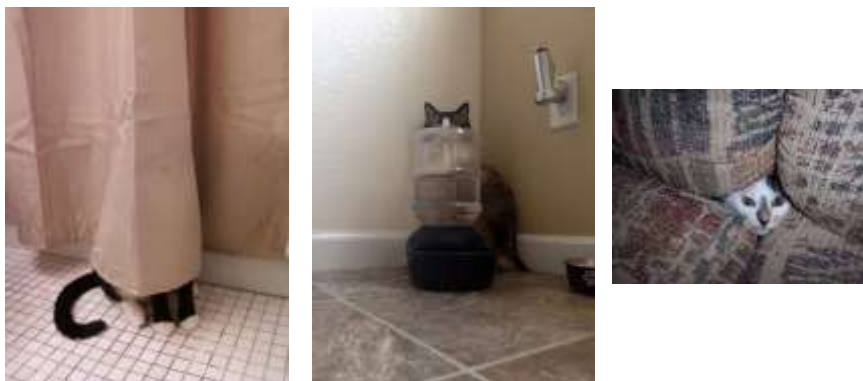
Challenges: Deformation



مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی **Keras**
علیرضا اخوان پور



Challenges: Occlusion



مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی **Keras**
علیرضا اخوان پور



Challenges: Background clutter



مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی **Keras**
علیرضا اخوان پور



Challenges: Intraclass variation



مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی **Keras**
علیرضا اخوان پور



An image classifier

```
def predict(image):
    # ???
    return class_label
```

Unlike e.g. sorting a list of numbers,

no obvious way to hard-code the algorithm for recognizing a cat, or other classes.



Data-driven approach:

1. Collect a dataset of images and labels
2. Use Machine Learning to train an image classifier
3. Evaluate the classifier on a withheld set of test images

```
def train(train_images, train_labels):
    # build a model for images -> labels...
    return model

def predict(model, test_images):
    # predict test_labels using the model...
    return test_labels
```



Supervised Learning (یادگیری نظارت شده)

- Supervised Learning uses **labeled** data to predict a label given some features.
- If the label is continuous its called a **regression** problem, if its categorical it is a **classification** problem.



دانشگاه گیلان

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

یادگیری نظارت شده – طبقه بندی

□ طبقه بندی یا classification:

▪ **feature** : قد و وزن

▪ **label** : جنسیت

□ هدف:

▪ تشخیص جنسیت از روی قد و وزن

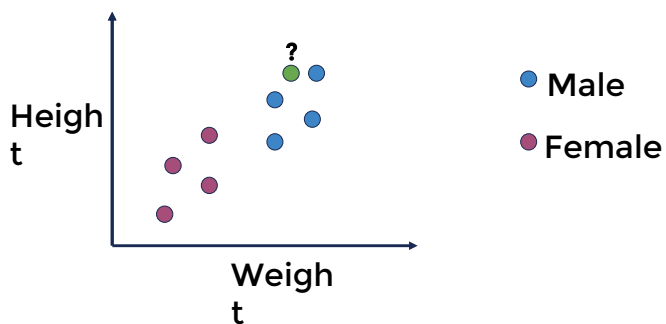


دانشگاه گیلان

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

یادگیری نظارت شده - طبقه بندی



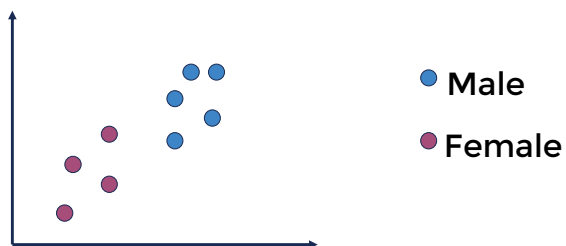
مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور



دانشگاه تهران

یادگیری نظارت شده - طبقه بندی



مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

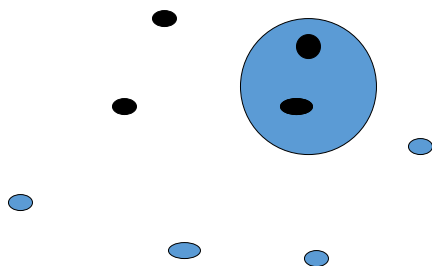
علیرضا اخوان پور



دانشگاه تهران

طبقه بندی: الگوریتم نزدیک ترین همسایه

□ 1-Nearest Neighbor



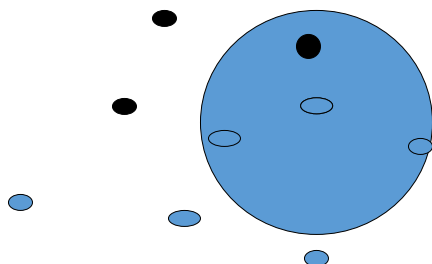
مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور



طبقه بندی: الگوریتم نزدیک ترین همسایه

□ 3-Nearest Neighbor

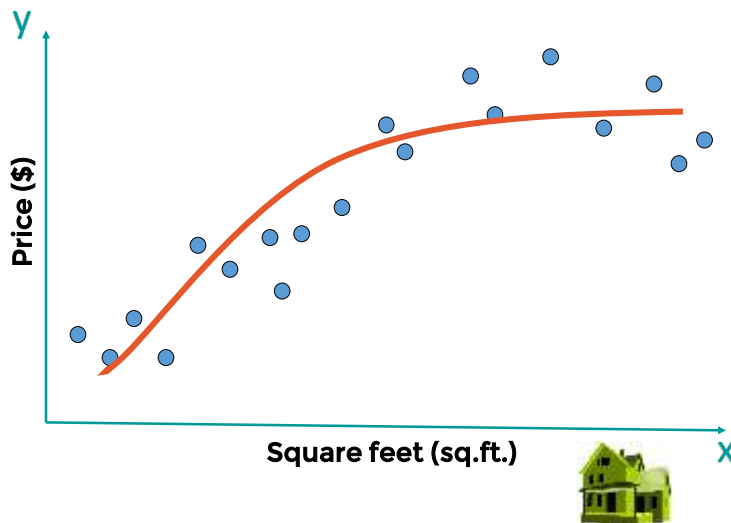


مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور



رگرسیون



مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور



مقدمه ۲:

تصویر به عنوان داده

Images as data

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور



تصویر به عنوان داده

```
import matplotlib.pyplot as plt
data = plt.imread('./image/sign.jpg')
plt.imshow(data)
plt.show()
```



مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی **Keras**
علیرضا اخوان پور



تصویر به عنوان داده

```
data.shape
(2832, 4256, 3)
```

```
data[1000, 1100]
array([189, 28, 43], dtype=uint8)
```

```
data[250, 3500]
array([ 66, 112, 197], dtype=uint8)
```



مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی **Keras**
علیرضا اخوان پور



آشنایی با تصویر در پایتون



01_Intro2image&numpy.ipynb

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



مقدمه ۳:

طبقه بندی تصویر در پایتون

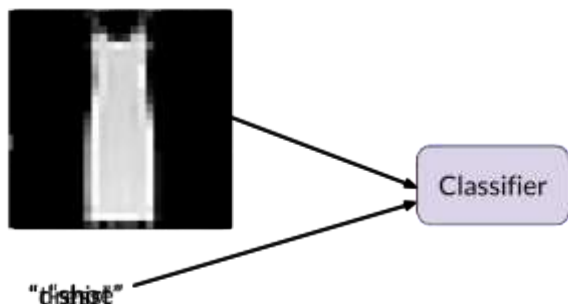
Image classification in Python

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



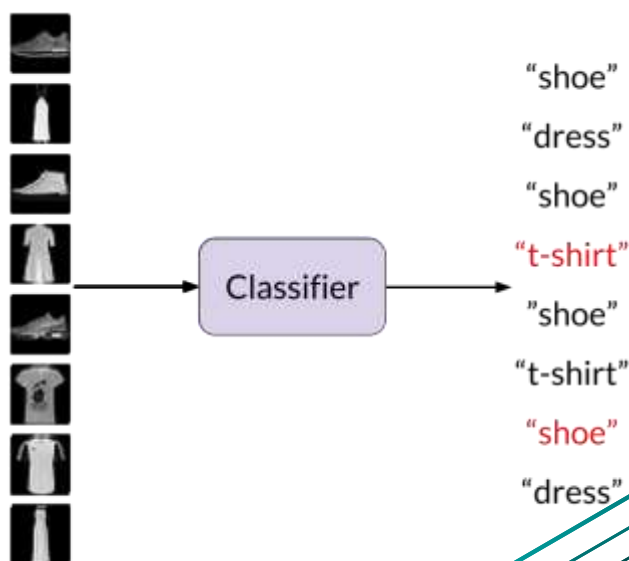
طبقه‌بندی تصویر: آموزش (Training)



مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور



طبقه‌بندی تصویر: ارزیابی (Evaluation)



Representing class data: one-hot encoding

```
labels = ["shoe", "dress", "shoe", "t-shirt",
          "shoe", "t-shirt", "shoe", "dress"]
```

	t-shirt	dress	shoe	
array([[0., 0., 1.]				<= shoe
[0., 1., 0.]				<= dress
[0., 0., 1.]				<= shoe
[1., 0., 0.]				<= t-shirt
[0., 0., 1.]				<= shoe
[1., 0., 0.]				<= t-shirt
[0., 0., 1.]				<= shoe
[0., 1., 0.]				<= dress

Keras مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی

علیرضا اخوان پور



One-hot encoding: testing predictions

Test

```
array([[0., 0., 1.],
       [0., 1., 0.],
       [0., 0., 1.],
       [0., 1., 0.],
       [0., 0., 1.],
       [0., 0., 1.],
       [0., 0., 1.],
       [0., 1., 0.]])
```

prediction

```
array([[0., 0., 1.],
       [0., 1., 0.],
       [0., 0., 1.],
       [1., 0., 0.] ✗,
       [0., 0., 1.],
       [1., 0., 0.] ✗,
       [0., 0., 1.],
       [0., 1., 0.]])
```

```
(test * prediction).sum()
```

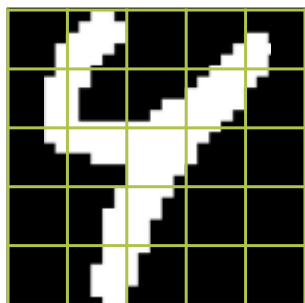
=6.0

Keras مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی

علیرضا اخوان پور



طبقه‌بندی تصویر:



`X = resize(img, (5, 5))`

`X.reshape(25)`

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



معرفی مجموعه داده هدی و تغییر اندازه و شکل برای استفاده در
الگوریتم‌های یادگیری ماشین



02_HodaDataset.ipynb

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



طبقه بندی نزدیکترین همسایه و K نزدیکترین همسایه در scikit-learn



03_K-Nearest Neighbor classification.ipynb

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



k-Nearest Neighbor on images **never used**.

- distance metrics on level of whole images can be very unintuitive



(all 3 images have same L2 distance to the one on the left)

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



مقدمه‌ای بر شبکه‌های عصبی

- Neurons and Activation Functions
- Cost Functions
- Gradient Descent
- Backpropagation

Keras مقدمه‌ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی

علیرضا اخوان پور



بخش اول:

مقدمه‌ای بر پرسپترون

Introduction to the Perceptron

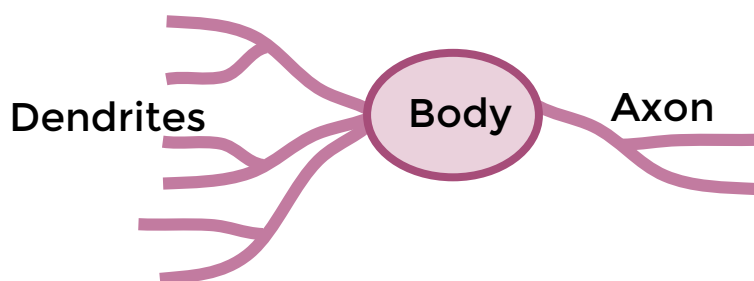
Keras مقدمه‌ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی

علیرضا اخوان پور



پرسپترون

□ The biological neuron:



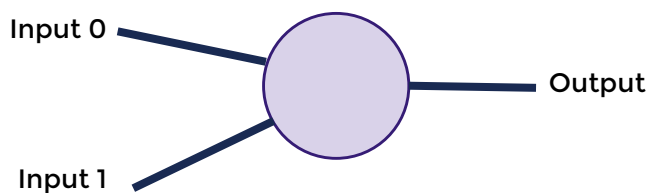
مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



پرسپترون

- نرون‌های مصنوعی هم ورودی و خروجی دارند
- این مدل ساده را پرسپترون می‌نامند.



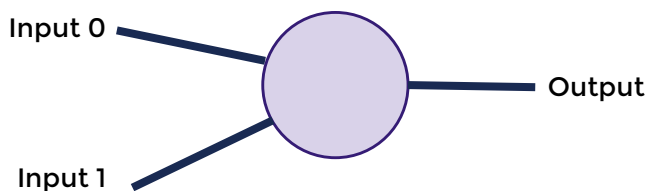
مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



پرسپترون

□ ورودی مقادیر ویژگی‌ها (features) هستند

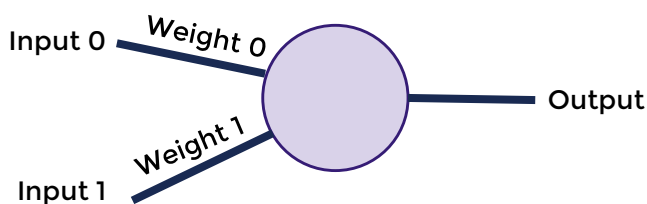


مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور



پرسپترون

□ ورودی‌ها در وزن‌ها ضرب می‌شوند

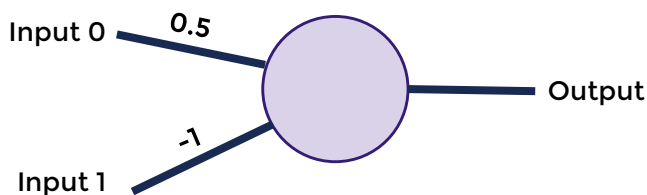


مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور



پرسپترون

□ وزن‌ها به صورت تصادفی و رندم مقداردهی اولیه می‌شوند

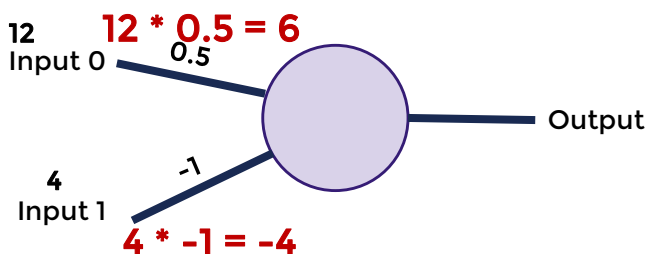


مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور



پرسپترون

□ ورودی‌ها در وزن‌های متناظر ضرب می‌شوند

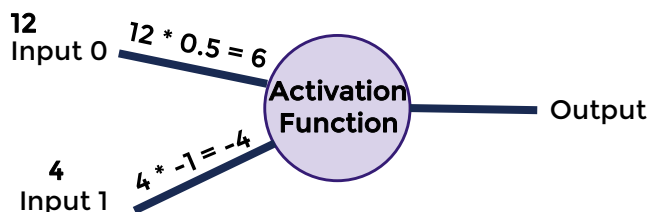


مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور



پرسپترون

□ یک تابع فعالیت بر روی نتایج اعمال می گردد



مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras
علیرضا اخوان پور

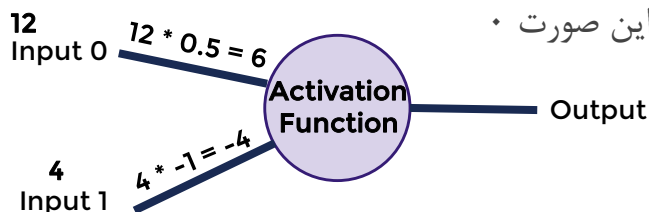


پرسپترون

□ انتخاب های زیادی برای تابع فعالیت داریم

□ ساده ترین تابع فعالیت:

- اگر مجموع مثبت بود : ۱
- در غیر این صورت ۰



مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras
علیرضا اخوان پور

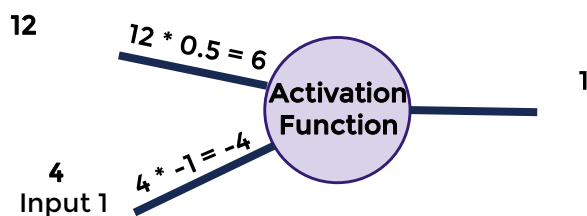


پرسپترون

□ در این مثال داریم:

$$6 - 4 = 2$$

پس خروجی برابر است با ۱



دانشگاه تهران

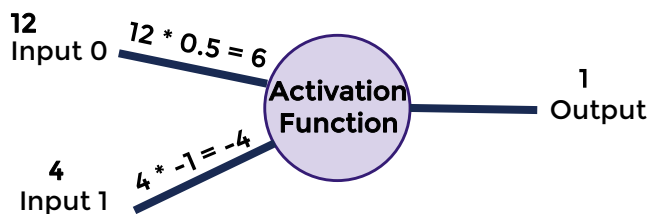
مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

پرسپترون

□ یک مشکل! اگر ورودی ۰ باشد؟!

□ مستقل از وزن‌ها خروجی ۰ می‌شود!



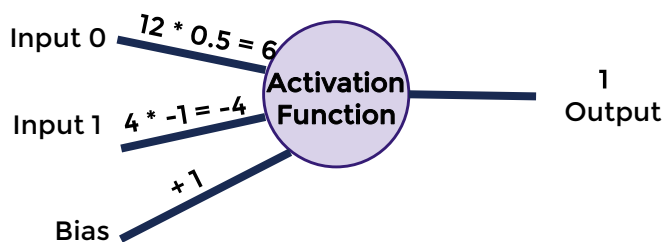
دانشگاه تهران

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

پرسپترون

□ این مشکل را با افزودن بایاس رفع می کنیم.

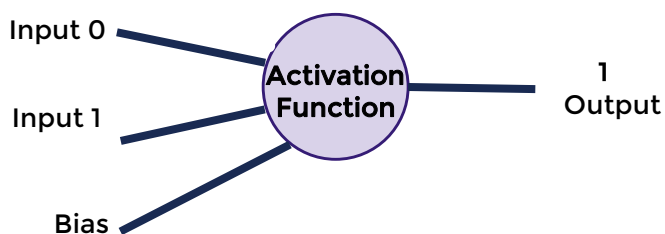


مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras
علیرضا اخوان پور



پرسپترون

□ اگر بخواهیم پرسپترون را به صورت ریاضی بنویسیم...



$$\sum_{i=0}^n w_i x_i + b$$

مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras
علیرضا اخوان پور



آنچه در بخش اول دیدیم...

- Biological Neuron
- Perceptron Model
- Mathematical Representation

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



بخش دوم:

مقدمه‌ای بر شبکه‌های عصبی

Introduction to Neural Networks

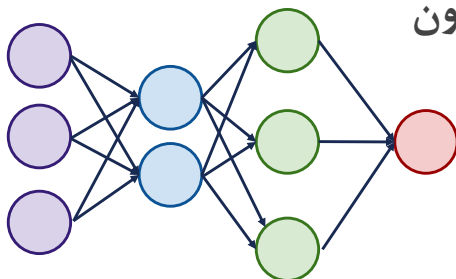
مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



مقدمه‌ای بر شبکه‌های عصبی

□ شبکه‌ای از چندین پرسپترون



Input Layer.

2 hidden layers.

Output Layer

مقدمه‌ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



دانشگاه تهران

مقدمه‌ای بر شبکه‌های عصبی - لایه‌ها

□ لایه ورودی:

- مقادیر حقیقی داده‌ها

□ لایه پنهان:

- لایه‌های بین ورودی و خروجی
- به ۳ یا بیشتر لایه معمولاً شبکه عمیق می‌گوییم.

□ لایه خروجی:

- تخمین نهایی شبکه از خروجی است

مقدمه‌ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

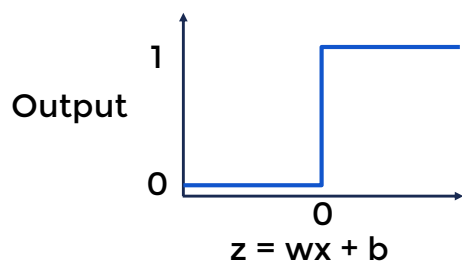
علیرضا اخوان پور



دانشگاه تهران

مقدمه‌ای بر شبکه‌های عصبی - تابع فعالیت

□ تابع فعالیت ساده‌ای که تا کنون دیدیم



□ تغییرات کوچک منعکس نمی‌شوند.



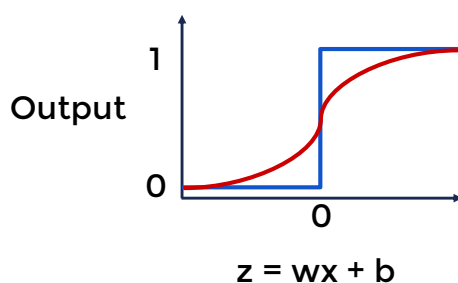
دانشگاه تهران

مقدمه‌ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

مقدمه‌ای بر شبکه‌های عصبی - تابع فعالیت

□ اگر تابعی پویاتر داشتیم بهتر میشد، مثل تابع قرمز رنگ!



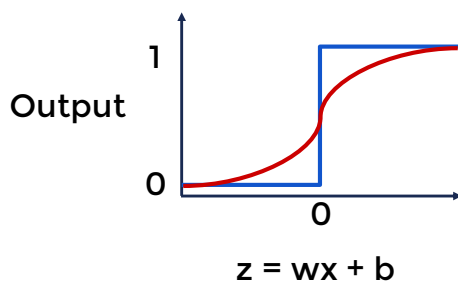
دانشگاه تهران

مقدمه‌ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

مقدمه‌ای بر شبکه‌های عصبی - تابع فعالیت

□ تابع سیگموئید



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



دانشگاه تهران

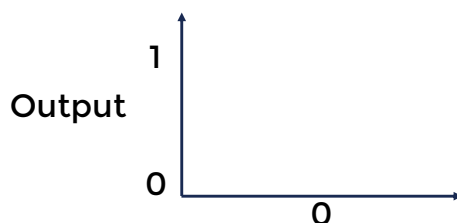
مقدمه‌ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

مقدمه‌ای بر شبکه‌های عصبی - تابع فعالیت

□ بسته به کار ممکن است از توابع فعالیت متفاوتی استفاده کنیم

□ توابع فعالیتی که ممکن است با آنها رو به رو شویم را ببینیم ...



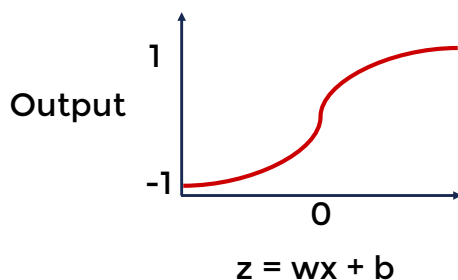
دانشگاه تهران

مقدمه‌ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

مقدمه‌ای بر شبکه‌های عصبی - تابع فعالیت

- Hyperbolic Tangent: $\tanh(z)$



$$\cosh x = \frac{e^x + e^{-x}}{2}$$

$$\sinh x = \frac{e^x - e^{-x}}{2}$$

$$\tanh x = \frac{\sinh x}{\cosh x}$$

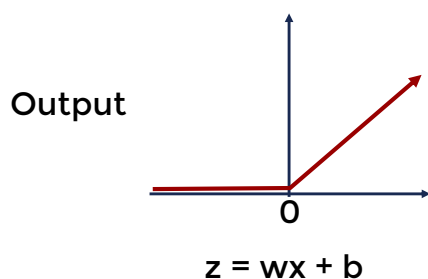
مقدمه‌ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور



مقدمه‌ای بر شبکه‌های عصبی - تابع فعالیت

ReLU □ - واحد یکسوساز خطی

- Rectified Linear Unit



$$\max(0, z)$$

مقدمه‌ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور



بخش سوم:

تابع هزینه

Cost Function



دانشگاه تهران

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

تابع هزینه

□ معیاری برای اندازه گیری بد بودن مدل!

بد بودن؟! تفاوت خروجی مدل با خروجی مورد انتظار

□ از علائم و متغیرهای زیر استفاده میکنیم:

Y' : نمایانگر خروجی مورد انتظار (true value)

Y : تخمین حاصل از خروجی شبکه

□ محاسبه a از وزن‌ها و بایاس

○ $w * x + b = z$

○ $\sigma(z) = y \Rightarrow$ Pass z into **activation function**



دانشگاه تهران

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

تابع هزینه

□ تابع هزینه ساده برای طبقه‌بندی:

$$C = \sum (y' - y)^2 / n$$

□ خطاهای بزرگتر به علت توان ۲، برجسته تر هستند.

□ این تابع خطا فرآیند آموزش را کند می‌کند.



دانشگاه تهران

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

تابع هزینه

□ Cross Entropy

$$C = - \sum_i y'_i \log(y_i)$$

□ این تابع اجازه می‌دهد که آموزش تسریع شود.

□ هر چه تفاوت بزرگتر باشد، سریعتر نورون می‌تواند یاد بگیرد.



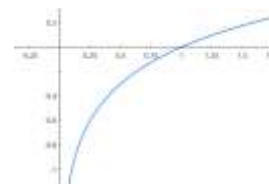
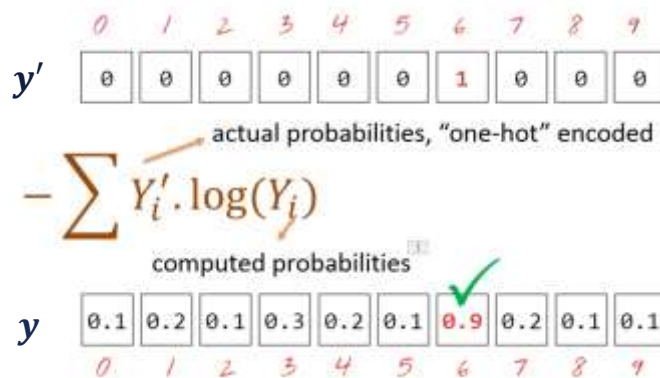
دانشگاه تهران

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

تابع هزینه

Cross Entropy □



Log (x)



دانشکده مهندسی کامپیوتر

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

تابع هزینه

□ تا الان با ۲ جنبه کلیدی یادگیری شبکه‌های عصبی آشنا شدیم:

✓ نورون‌ها با تابع فعالیت

✓ تابع هزینه

□ ما هنوز یک گام کلیدی را کم داریم! "یادگیری!"



دانشکده مهندسی کامپیوتر

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور

بخش چهارم:

پس انتشار خطا و گرادیان کاهشی

Gradient Descent and Backpropagation



دانشگاه تهران

مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور

پس انتشار خطا و گرادیان کاهشی

□ گرادیان کاهشی یا Gradient descent

➤ یک الگوریتم بهینه سازی برای یافتن حداقل یک تابع است.

□ برای یافتن کمینه ی محلی یک تابع با استفاده از این الگوریتم،
گام هایی متناسب با منفی گرادیان تابع در محل فعلی برداشته خواهد شد.



دانشگاه تهران

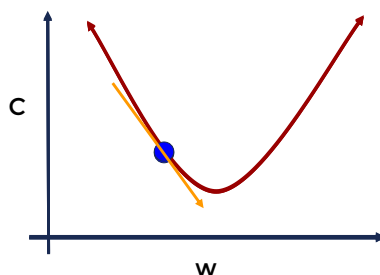
مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور

پس انتشار خطا و گرادیان کاهش

□ گرادیان کاهش (در یک بعد)

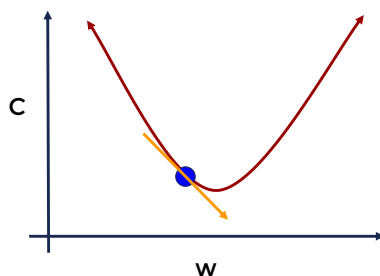
□ Gradient Descent (in 1 dimension)



پس انتشار خطا و گرادیان کاهش

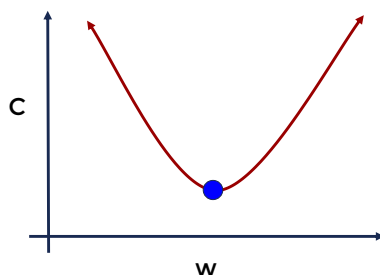
□ گرادیان کاهش (در یک بعد)

□ Gradient Descent (in 1 dimension)



پس انتشار خطا و گرادیان کاهش

□ به صورت بصری می توانیم مقدار انتخابی برای به حداقل رساندن هزینه را ببینیم!



مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور



پس انتشار خطا و گرادیان کاهش

□ پیدا کردن این کمینه برای ۱ بعد ساده است، اما تقریباً تمام مسائل ما دارای پارامترهای زیادی است، به این معنی که ما باید از جبر خطی built-in در کتابخانه یادگیری عمیق استفاده کنیم.

□ با استفاده از گرادیان کاهش می توانیم بهترین پارامترها را برای به حداقل رساندن تابع هزینه بدست آوریم، به عنوان مثال، پیدا کردن بهترین مقادیر برای وزن ورودی های نورون.

مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور



پس انتشار خطا و گرادیان کاهشی

□ حالا ما فقط یک مسئله برای حل کردن داریم، چگونه می توانیم پارامترهای مطلوب یا وزن ها را در کل شبکه تنظیم کنیم؟

□ اینجاست که پس انتشار خطا یا **backpropagation** مطرح می شود.



دانشگاه تهران

مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور

پس انتشار خطا و گرادیان کاهشی

□ پس انتشار خطا برای محاسبه سهم خطای هر نورون پس از پردازش یک دسته از داده ها استفاده می شود.

□ این روش به شدت بر قاعده زنجیری برای پیدا کردن مشتق یا **chain rule** تکیه می کند تا از طریق شبکه عقب برود و این خطاها را محاسبه کند.



دانشگاه تهران

مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور

پس انتشار خطا و گرادیان کاهشی

□ پس انتشار خطا با محاسبه خطای خروجی و با بازگشت لایه به لایه از خروجی به سمت ورودی انجام می دهد.

□ برای هر مقدار ورودی نیاز به یک خروجی مورد انتظار دارد (یادگیری نظارت شده).



انستیتو ملی اطلاعات علمی و فنی

مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور

playground.tensorflow.org



انستیتو ملی اطلاعات علمی و فنی

مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور

بخش پنجم:

پیاده سازی در کراس

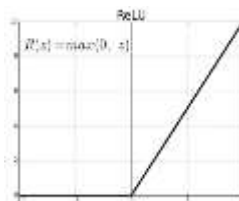
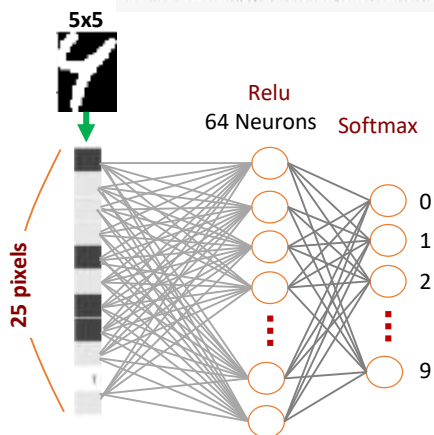
Implementation with Keras!

مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras
علیرضا اخوان پور



پیاده سازی در کراس - معماری مدل

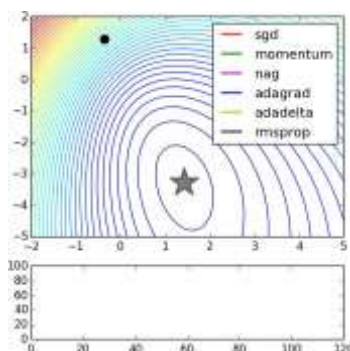
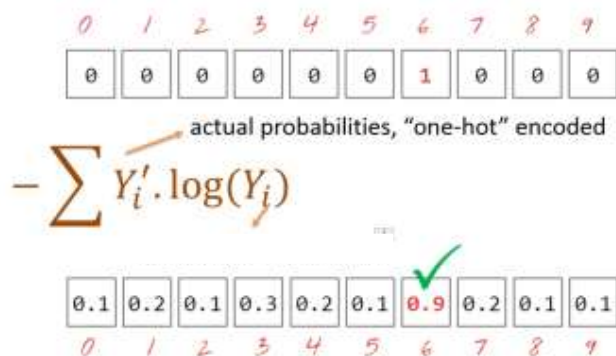
```
model = Sequential()
model.add(Dense(64, activation='relu', input_dim=25))
model.add(Dense(10, activation='softmax'))
```



$$\text{softmax}(L_n) = \frac{e^{L_n}}{\|e^L\|}$$

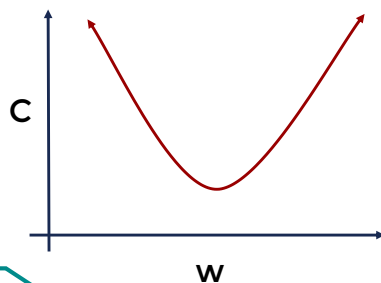
پیاده سازی در کراس - تابع هزینه و بهینه سازی

```
model.compile(loss='categorical_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])
```



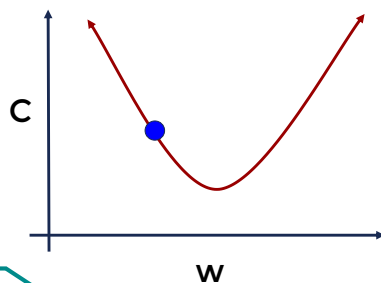
پیاده سازی در کراس: تعیین داده آموزشی، اندازه mini-batch و تعداد epoch

```
model.fit(x_train, y_train,
          epochs=30,
          batch_size=64)
```



پیاده سازی در کراس: تعیین داده آموزشی، اندازه mini-batch و تعداد epoch

```
model.fit(x_train, y_train,
          epochs=30,
          batch_size=64)
```

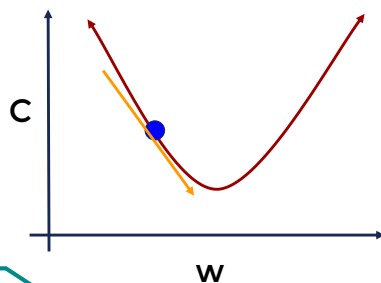


مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras
علیرضا اخوان پور



پیاده سازی در کراس: تعیین داده آموزشی، اندازه mini-batch و تعداد epoch

```
model.fit(x_train, y_train,
          epochs=30,
          batch_size=64)
```

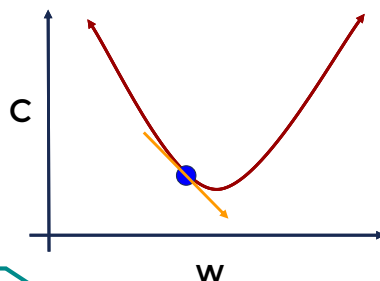


مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras
علیرضا اخوان پور



پیاده سازی در کراس: تعیین داده آموزشی، اندازه mini-batch و تعداد epoch

```
model.fit(x_train, y_train,
          epochs=30,
          batch_size=64)
```

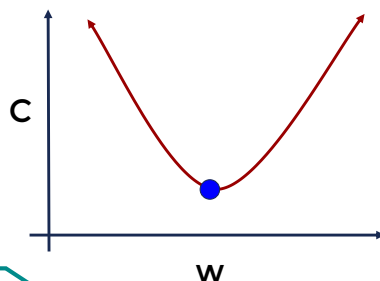


مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras
علیرضا اخوان پور



پیاده سازی در کراس: تعیین داده آموزشی، اندازه mini-batch و تعداد epoch

```
model.fit(x_train, y_train,
          epochs=30,
          batch_size=64)
```



مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras
علیرضا اخوان پور



پیاده سازی شبکه عصبی در کراس - ارقام فارسی



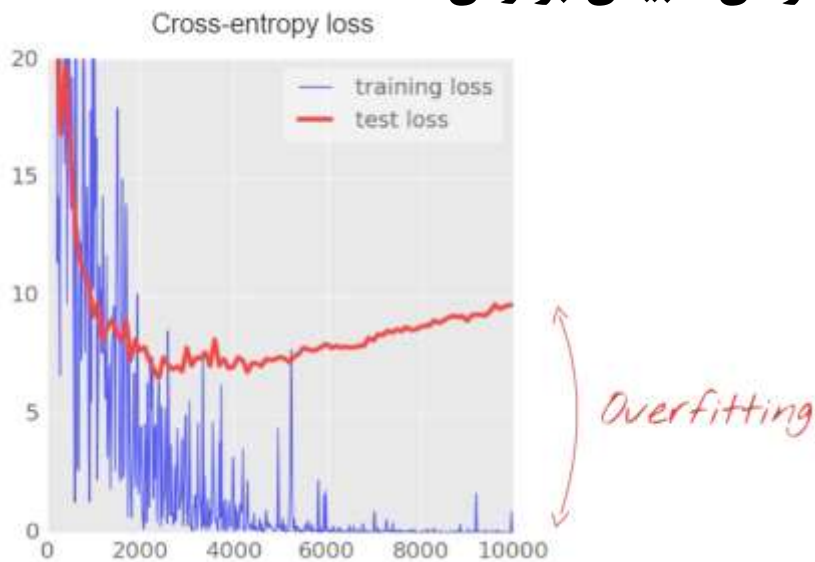
04_a Gentle Introduction to Keras - Simple neural network(mlp).ipynb

مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

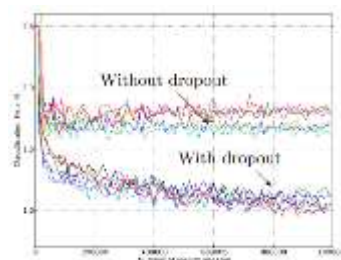
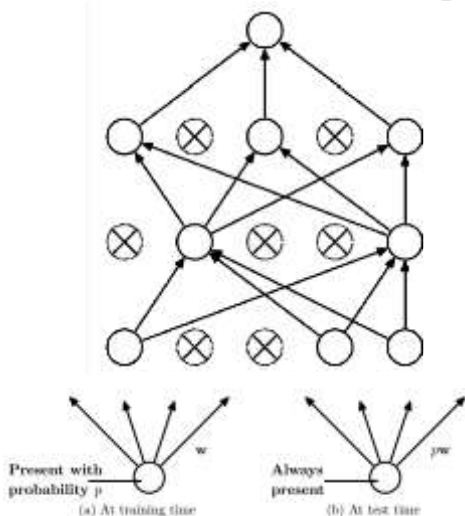
علیرضا اخوان پور



پیاده سازی در کراس - بیش برازش



پیاده سازی در کراس - Dropout



Main Idea: approximately combining exponentially many different neural network architectures efficiently

Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, JMLR (2014)

بیش برازش داده‌ها و لایه‌ی Dropout در کراس



05_Dropout.ipynb

بخش ششم:

شبکه های عصبی کانولوشنالی

Convolutional Neural Networks



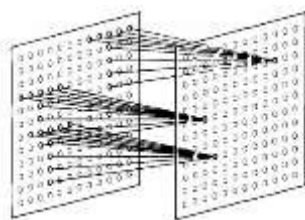
دانشگاه تهران

مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور

A bit of history:

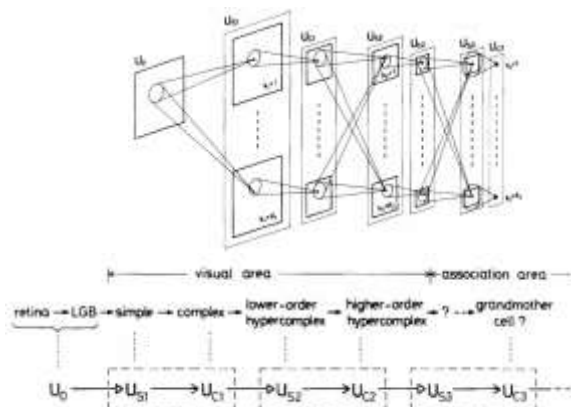
Neurocognitron
[Fukushima 1980]



"sandwich" architecture (SCSCSC...)

simple cells: modifiable parameters

complex cells: perform pooling



دانشگاه تهران

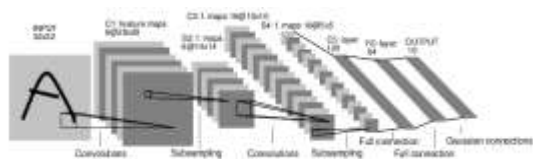
مقدمه ای بر شبکه های عصبی و پیاده سازی در کتابخانه Keras

علیرضا اخوان پور

A bit of history:

Gradient-based learning applied to document recognition

[LeCun, Bottou, Bengio, Haffner 1998]

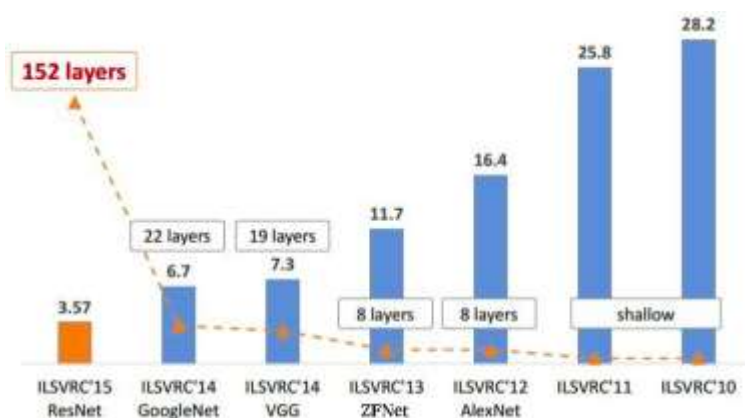


LeNet-5

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور



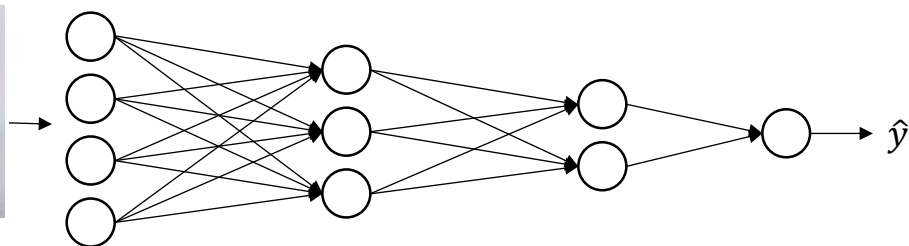
بعد از ۲۰۱۲...



مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور



شهود لزوم سلسله مراتب و عمق شبکه



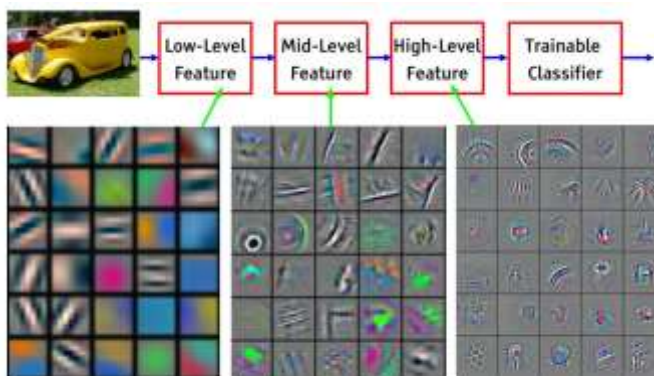
مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



شهود لزوم سلسله مراتب و عمق شبکه

[From recent Yann LeCun slides]



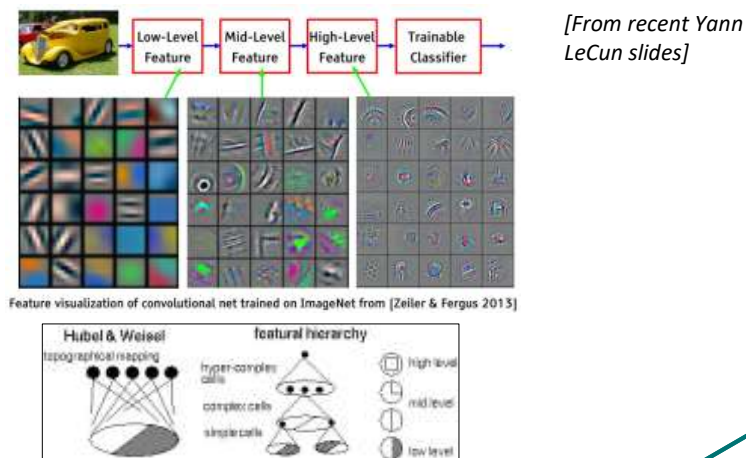
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras

علیرضا اخوان پور



شهود لزوم سلسله مراتب و عمق شبکه



Keras مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی
علیرضا اخوان پور



یادگیری عمیق و تصاویر بزرگ

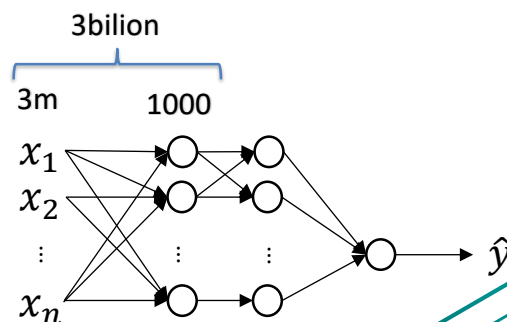


$64 \times 64 \times 3$
 $= 12288$



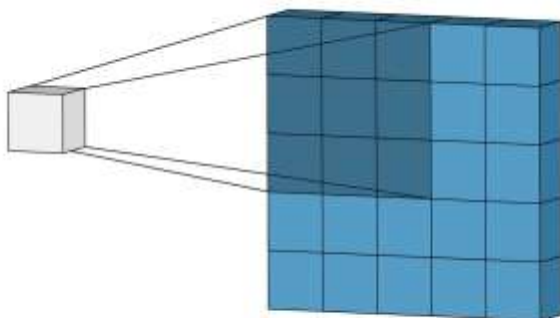
$1000 \times 1000 \times 3$
 $= 3000000 = 3m$

→ Cat? (0/1)



Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

مقدمات convolution – فیلتر edge detection



1	0	-1
1	0	-1
1	0	-1

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

مقدمات convolution – لبه‌یابی

3 ¹	0 ⁰	1 ⁻¹	2 ⁻¹	7 ⁻¹	4 ⁻¹
1 ¹	5 ⁰	8 ⁻¹	9 ⁻¹	3 ⁻¹	1 ⁻¹
2 ¹	7 ⁰	2 ⁻¹	5 ⁻¹	1 ⁻¹	3 ⁻¹
0 ¹	1 ⁰	3 ⁻¹	1 ⁻¹	7 ⁻¹	8 ⁻¹
4	2	1	6	2	8
2	4	5	2	3	9

*

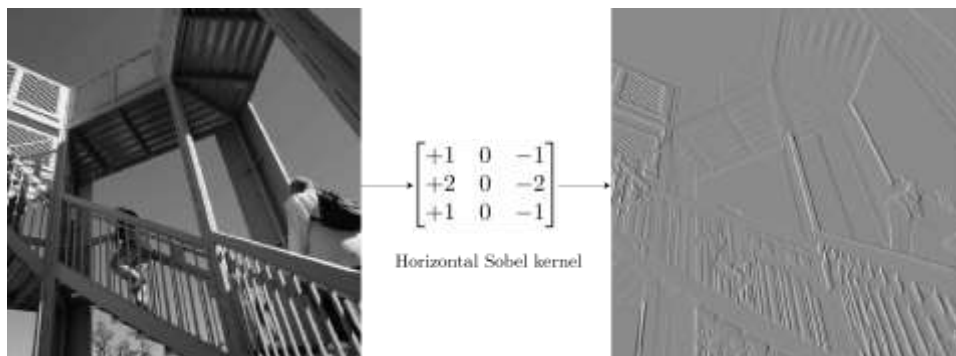
1	0	-1
1	0	-1
1	0	-1

=

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

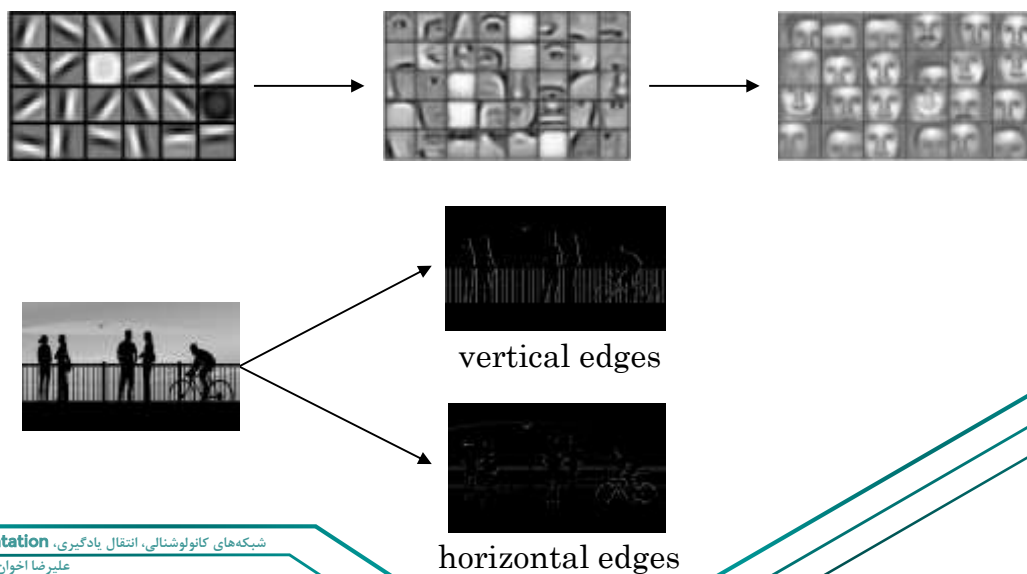
Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

مقدمات convolution



Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

مقدمات convolution – لبه‌یابی



Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

یادگیری فیلتر!

1	0	-1
1	0	-1
1	0	-1

1	0	-1
2	0	-2
1	0	-1

3	0	-3
10	0	-10
3	0	-3

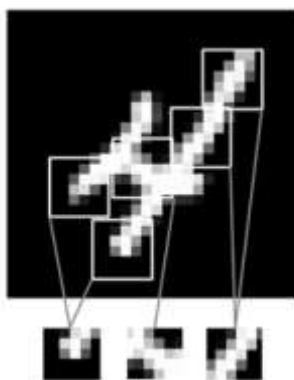
3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

45 ?
70?
73?

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

The convolution operation

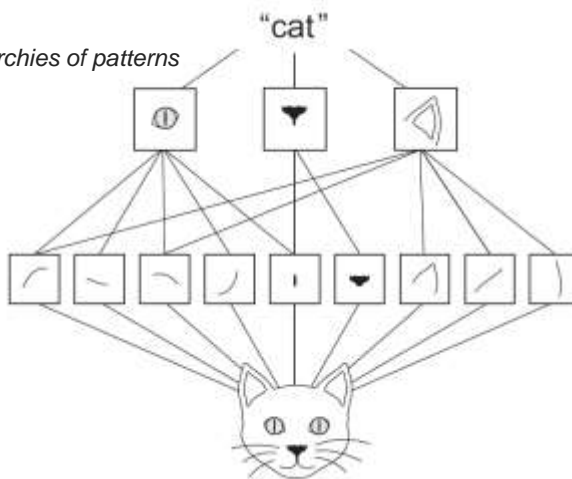


- Dense layers learn global patterns in their input feature space
- Convolution layers learn local patterns

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

The patterns they learn are translation invariant

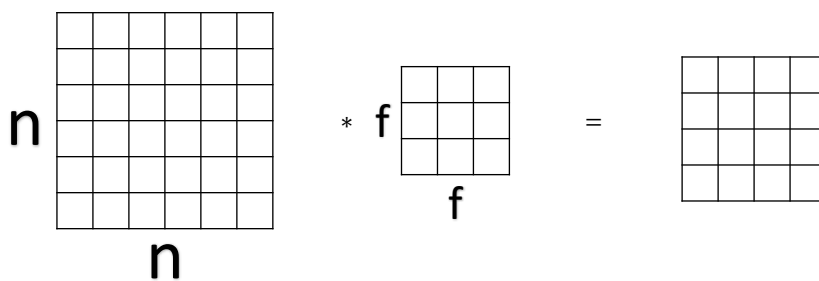
They can learn spatial hierarchies of patterns



Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

مشکلات کانولوشن!



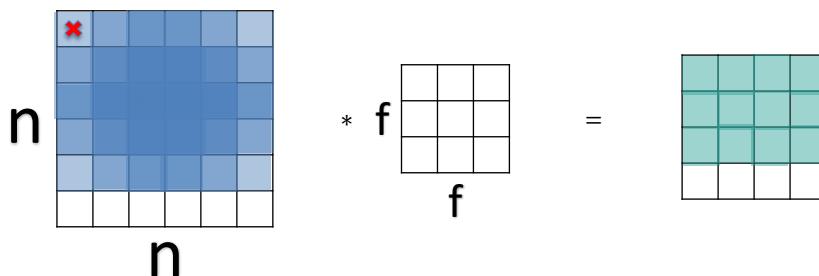
- لبه‌ها به اندازه ی پیکسل‌های درونی در کانولوشن شرکت نمیکنند
- خروجی کوچک میشود

$$n-f+1$$

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

مشکلات کانولوشن!



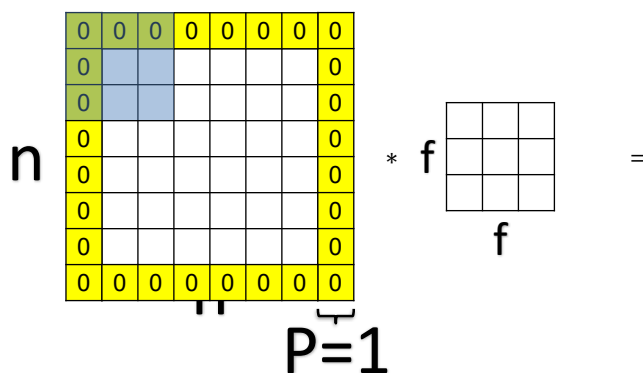
- لبه ها به اندازه ی پیکسل های درونی در کانولوشن شرکت نمیکنند
- خروجی کوچک میشود

$$n-f+1$$

Data augmentation شبکه های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

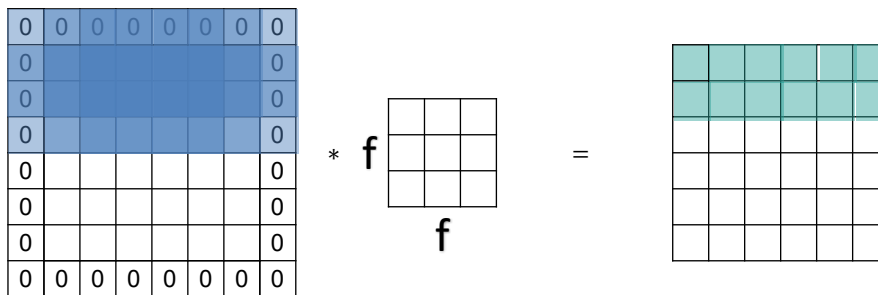
راه کار؟ padding



Data augmentation شبکه های کانولوشنالی، انتقال یادگیری،

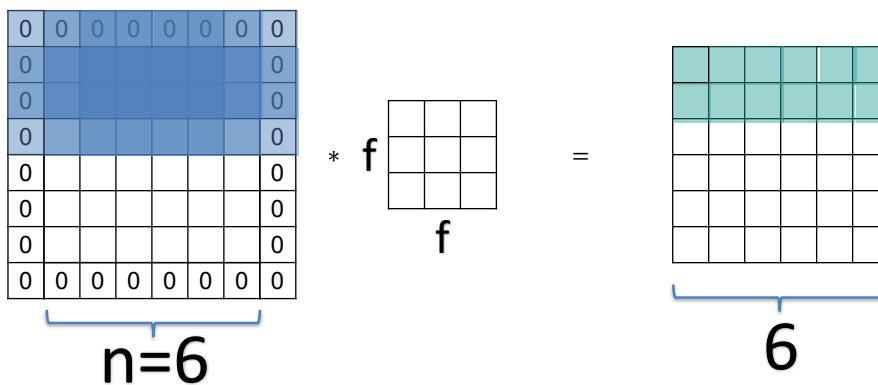
علیرضا اخوان پور

padding



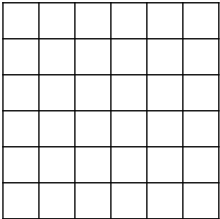
Data augmentation، انتقال یادگیری، شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

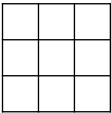
padding



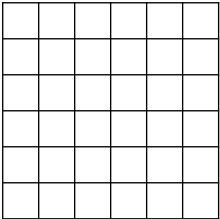
Data augmentation، انتقال یادگیری، شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

padding

n

 n

$*$ f

 f

$=$



$P = \text{padding}$

$n+2p-f+1$

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری، علیرضا اخوان پور

کانولوشن Valid و Same

“Valid”:

$$n \times n * f \times f \rightarrow n-f+1 \times n-f+1$$

$$6 \times 6 * 3 \times 3 \rightarrow 4 \times 4$$

“Same”:

Pad so that output size is the same as the input size.

$$n+2p-f+1 \times n+2p-f+1$$

$$\cancel{n+2p-f+1} = \cancel{n} \Rightarrow p = \frac{f-1}{2}$$

برای مثال قبل $p=1$

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری، علیرضا اخوان پور

Stride در کانولوشن

2	3	4	7	3	4	6	3	4	9	4
6	1	6	0	9	1	8	0	7	1	4
3	3	4	3	3	3	4	3	9	4	7
7	1	8	0	3	1	6	0	6	1	3
4	3	2	4	1	3	8	4	3	4	6
3	1	2	0	4	1	0	9	1	8	0
0	1	0	3	1	9	0	2	1	0	4

 $*$

3	4	4
1	0	2
-1	0	3

 $=$

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

خلاصه کانولوشن و خروجی آن

$n \times n$ image $f \times f$ filter

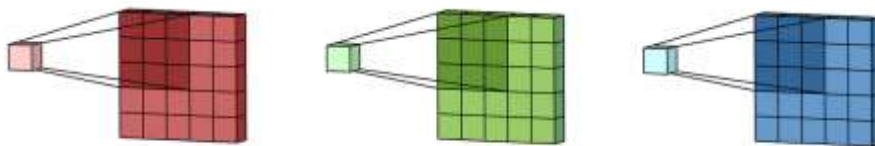
padding p stride s

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

کانولوشن بر روی تصویر رنگی (RGB)



به ازای هر کانال رنگی یک فیلتر

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

کانولوشن بر روی تصویر رنگی (RGB)



جمع نظیر به نظیر feature map یا activation map

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

کانولوشن بر روی تصویر رنگی (RGB)

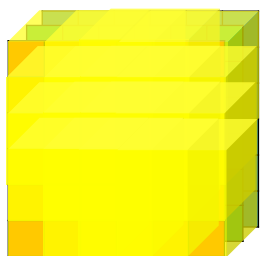


+ بایاس

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

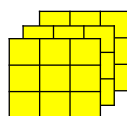
علیرضا اخوان پور

کانولوشن بر روی تصویر رنگی (RGB)



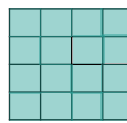
6 x 6 x 3

*



3 x 3

=

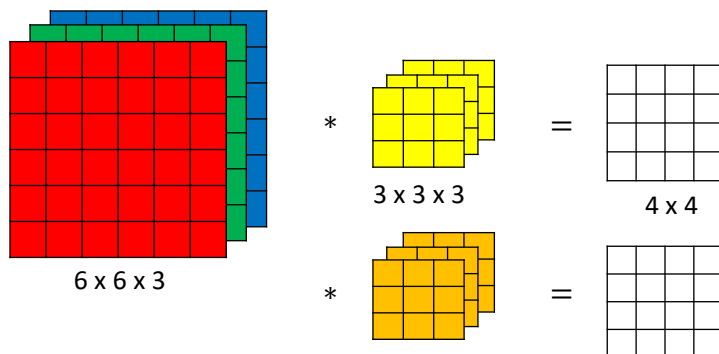


4 x 4

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

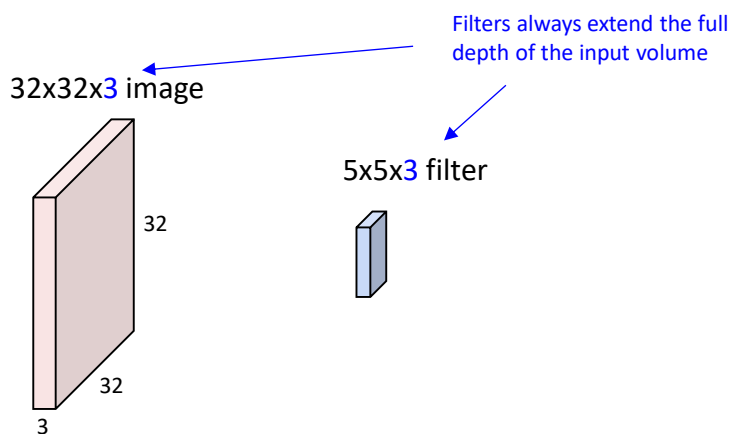
چندین فیلتر...



Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

چندین فیلتر...

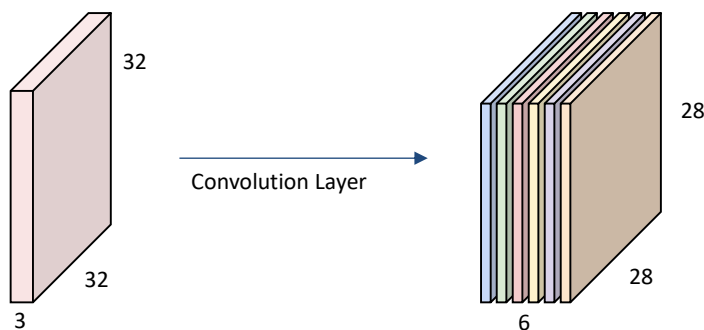


Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

چندین فیلتر...

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a "new image" of size 28x28x6!

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

محاسبه تعداد پارامتر

Number of parameters in one layer

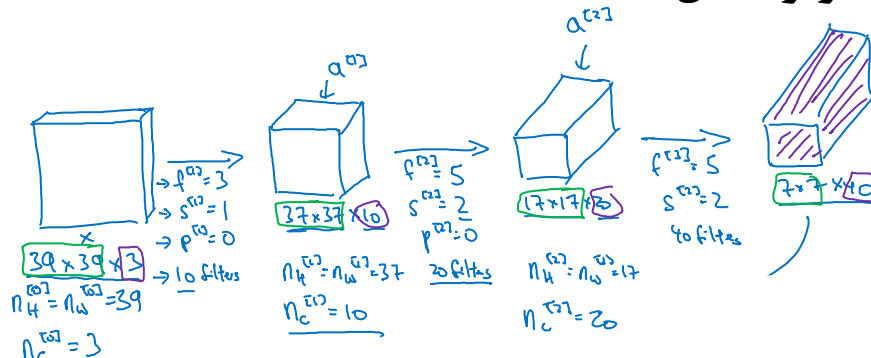
If you have 10 filters that are 3 x 3 x 3 in one layer of a neural network, how many parameters does that layer have?

تعداد پارامترهای هر فیلتر $3 \times 3 \times 3 + 1 = 28$

$28 \times 10 = 280$

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

یک شبکه کانولوشنال ساده



$$\frac{n+2p-f}{s} + 1$$

$$\frac{39+0-3}{1} + 1 = 37$$

Data augmentation, شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

انواع لایه در شبکه‌های کانولوشنالی

- ☐ Convolution (Conv)
- ☐ Pooling (Pool)
- ☐ Fully Connected (FC)

Data augmentation, شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

لایه‌ی pooling

1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

9	2
6	3

☐ دو hyper parameter دارد: s و f
☐ هیچ پارامتری برای آموزش ندارد

شبکه‌های کانولوشنالی، انتقال یادگیری، Data augmentation

علیرضا اخوان پور

لایه‌ی max pooling – pooling

$5 \times 5 \times 2$

1	3	2	1	3
2	9			5
1				2
8	3		1	0
5	6	1	2	9

5×5

$f = 3$

$s = 1$

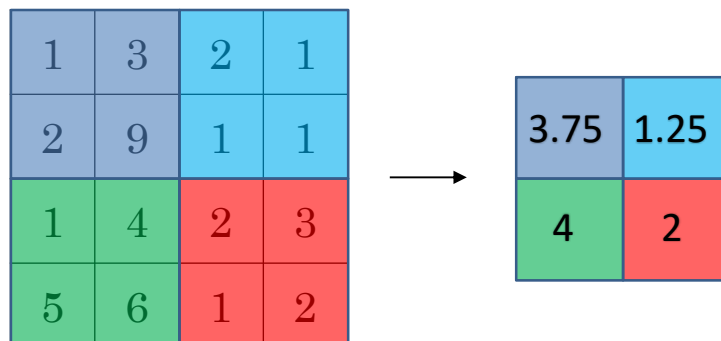
$3 \times 3 \times 2$

9	9	5
9	9	5
8	6	9

شبکه‌های کانولوشنالی، انتقال یادگیری، Data augmentation

علیرضا اخوان پور

Average pooling



شبکه‌های کانولوشنالی، انتقال یادگیری، Data augmentation

علیرضا اخوان پور

خلاصه pooling

Hyperparameters:

f : filter size

s : stride

Max or average pooling

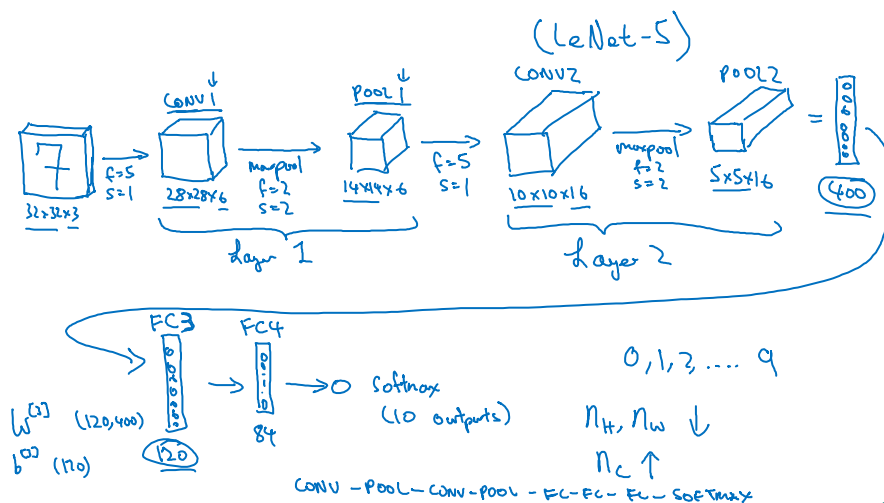
$f = 2, s = 2$

$f = 3, s = 2$

شبکه‌های کانولوشنالی، انتقال یادگیری، Data augmentation

علیرضا اخوان پور

مثالی از یک شبکه کانولوشنالی (LeNet-5)



Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

سایز هر لایه و تعداد پارامترها

	Activation shape	Activation Size	# parameters*
Input:	(32,32,3)	3,072	0
CONV1 ($f=5, s=1$)	(28,28,8)	6,272	608
POOL1	(14,14,8)	1,568	0
CONV2 ($f=5, s=1$)	(10,10,16)	1,600	3216
POOL2	(5,5,16)	400	0
FC3	(120,1)	120	48, 120
FC4	(84,1)	84	10,164
Softmax	(10,1)	10	850

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

```

model = Sequential()
model.add(layers.Conv2D(8, (5, 5), activation='relu',
                        input_shape=(32, 32, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(16, (5, 5), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(120, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(84, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))

```

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

```
model.summary()
```

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 28, 28, 8)	608
max_pooling2d_11 (MaxPooling)	(None, 14, 14, 8)	0
conv2d_13 (Conv2D)	(None, 10, 10, 16)	3216
max_pooling2d_12 (MaxPooling)	(None, 5, 5, 16)	0
flatten_6 (Flatten)	(None, 400)	0
dense_12 (Dense)	(None, 120)	48120
dropout_6 (Dropout)	(None, 120)	0
dense_13 (Dense)	(None, 84)	10164
dense_14 (Dense)	(None, 10)	850
Total params: 62,958		
Trainable params: 62,958		
Non-trainable params: 0		

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،

علیرضا اخوان پور

شبکه‌های کانولوشنالی در کراس - ارقام فارسی



06_ConvolutionalNeuralNetwork-Hoda-Keras.ipynb

Data augmentation شبکه‌های کانولوشنالی، انتقال یادگیری،
علیرضا اخوان پور

منابع

- <http://cs231n.stanford.edu/>
- <https://www.csee.umbc.edu/courses/671/fall01/class-notes/k-nn.ppt>
- https://docs.google.com/presentation/d/1kytN5gqrZe0Bcu3hExjZpA4PQipaFEKn26S-JhqncXk/edit#slide=id.g1ff36e20db_0_450
- <https://www.slideshare.net/Alirezaakhavanpour/tensorflow-71395844>
- <https://www.datacamp.com/courses/convolutional-neural-networks-for-image-processing>



دانشگاه تهران

مقدمه ای بر شبکه‌های عصبی و پیاده‌سازی در کتابخانه‌ی Keras
علیرضا اخوان پور