یادگیری عمیق

دکتر امین گلزاری اسکوئی

a.golzari@azaruniv.ac.ir

a.golzari@tabrizu.ac.ir

https://github.com/Amin-Golzari-Oskouei

دانشگاه شهید مدنی آذربایجان

پاییز ۱۴۰۱

فصل ۳
شروع کار با شبکه‌های عصبی

مطالب این فصل

- مؤلفه‌های اصلی شبکه‌های عصبی
- مقدمه‌ای بر کراس
- نصب و راه‌اندازی محیط یادگیری عمیق
- استفاده از شبکه‌های عصبی برای حل مسئله‌های رگرسیون و دسته‌بندی

مرور پایتون

Enumerate

In [1]:
```
a = ['ali', 'hasan', 'hossein', 'reza']

for i,c in enumerate(a):
    print(i,c)
```

```
0 ali
1 hasan
2 hossein
3 reza
```

List comprehension

$s = \{x^2 : x \text{ in } \{0,...,9\}\}$

In [1]:
```
s = [x**2 for x in range(10)]
print(s)
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

In [2]:
```
arr = [[1,3,-1],
       [2,1,5],
       [-1,-2,-3]]
[max(a) for a in arr]
```

Out[2]:
```
[3, 5, -1]
```

مرور پایتون

Dictionary

In [1]:
```
dic = {'Tehran':21, 'Qom':25, 'Isfahan':31}
dic['Qom']
```

Out[1]: 25

In [2]:
```
for key, value in dic.items():
    print(key, value)
```

```
Tehran 21
Isfahan 31
Qom 25
```

In [3]:
```
dic.get('Qom', '?')
```

Out[3]: 25

In [4]:
```
dic.get('Tabriz', '?')
```

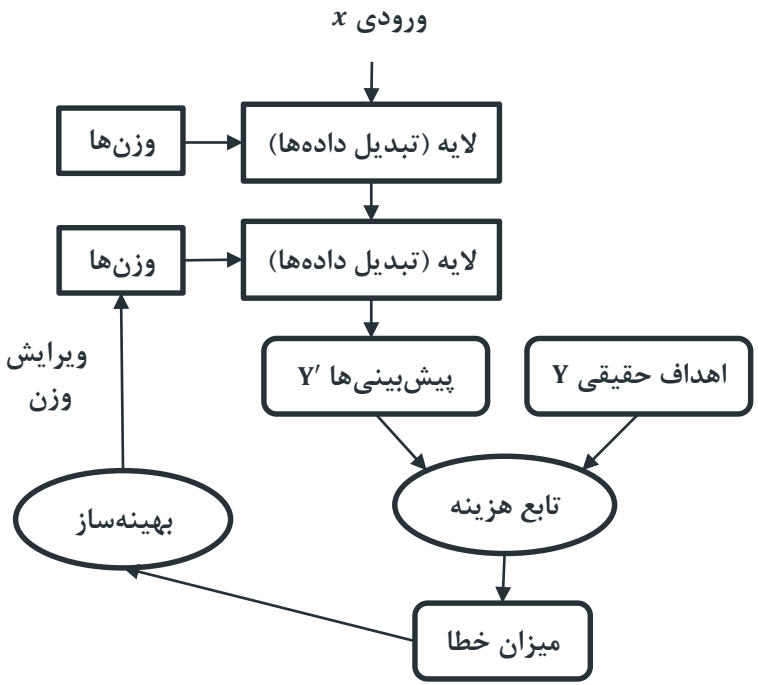
Out[4]: '?'

ساختار شبکه عصبی

همان‌طور که در فصل‌های قبلی مشاهده کردید، آموزش شبکه عصبی مبتنی بر اهداف زیر است:

- لایه‌ها که ترکیب آن‌ها شبکه (مدل) را به وجود می‌آورد.
- نمونه‌های ورودی و اهداف مرتبط با آن‌ها
- تابع هزینه، جهت ایجاد سیگنال بازخورد آموزش
- بهینه‌ساز که نحوه انجام یادگیری را تعیین می‌کند

ساختار شبکه عصبی



لایه‌ها: بلوک‌های سازنده یادگیری عمیق

در اینجا لایه‌ای ایجاد می‌کنیم که فقط تنسورهای دوبعدی را به عنوان ورودی قبول خواهد کرد و اولین بعد آن 784 خواهد بود (محور صفر مشخص‌کننده تعداد کل نمونه‌هاست که اشاره‌ای به آن نشده و از این رو هر مقداری پذیرفته خواهد شد). این لایه تنسوری را برمی‌گرداند که اولین بعدش 32 است.

```
from tensorflow.keras import layers

layer = layers.Dense(32, input_shape=(784,))
```

لایه‌ی متراکم با 32 خروجی

لایه‌ها: بلوک‌های سازنده یادگیری عمیق

بنابراین لایه بعدی بردارهای 32 بعدی را به عنوان ورودی قبول می‌کند. هنگام استفاده از کراس، نگرانی در مورد سازگاری لایه‌ها بی‌مورد است، چراکه لایه‌های جدید به نحوی اضافه می‌شوند که با شکل لایه قبلی خود همخوانی داشته باشند. به عنوان مثال، فرض کنید برنامه زیر را می‌نویسید:

```
from tensorflow.keras import models
from tensorflow.keras import layers


model = models.Seuential()
model.add(layers.Dense(32, input_shape=(784,)))
model.add(layers.Dense(32))
```

مدل‌ها: شبکه‌ای از لایه‌ها

- متداول‌ترین نمونه مدل‌ها، تعدادی لایه متوالی است که یک ورودی را به خروجی خاصی تبدیل می‌کند.
- هر چه جلوتر بروید با انواع بیشتری از توپولوژی‌های شبکه روبرو خواهید شد. برخی از انواع متداول به شرح زیر است:
  - شبکه‌های دوشاخه
  - شبکه‌های چندسر
  - بلوک‌های انسپشن
- انتخاب معماری صحیح شبکه بیشتر کار هنری است تا علمی؛ و با وجود اینکه اصول و تجارب ارزنده قابل‌اتکایی در این زمینه وجود دارند، فقط تمرین و تکرار می‌تواند از شما یک معمار شبکه عصبی بسازد. در چند فصل بعدی اصول ساخت شبکه‌های عصبی به طور کامل توضیح داده شده است که در انتخاب شبکه عصبی مناسب برای هر مسئله خاص به شما کمک می‌کنند.

توابع هزینه و بهینه‌سازها: دو عنصر کلیدی فرایند یادگیری

- بعد از انتخاب معماری شبکه، دو مؤلفه دیگر هم باید انتخاب شوند:

  - **تابع هزینه (تابع هدف):** کمیتی که در طول آموزش به حداقل خواهد رسید. تابع هزینه معیاری برای سنجش موفقیت در مسئله جاری است.

  - **بهینه‌ساز:** نحوه به‌روزرسانی شبکه بر اساس تابع هزینه را تعیین می‌کند. بهینه‌ساز، گونه‌ای از گرادیان نزولی تصادفی (SGD) را پیاده‌سازی می‌کند.

مقدمه‌ای بر کراس

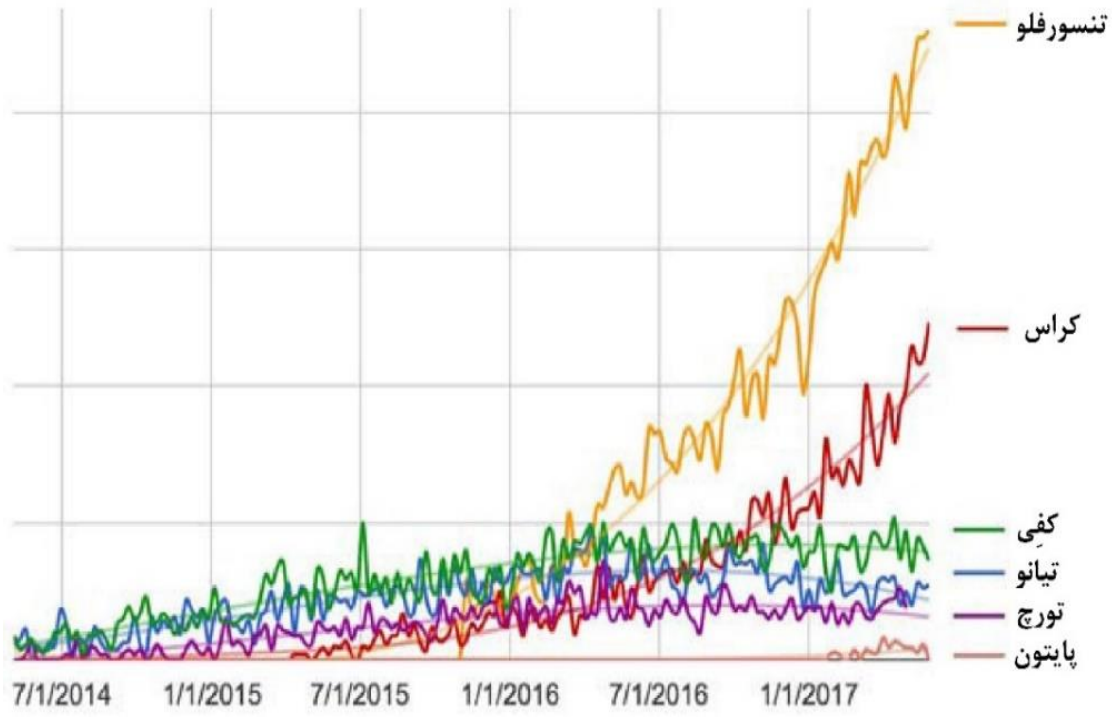
- کراسِ چارچوب یادگیری عمیق برای پایتون است که روشی ساده را برای تعریف و آموزش تقریباً هر نوع مدل یادگیری عمیق فراهم می‌آورد.

مقدمه‌ای بر کراس

ویژگی‌های کلیدی کراس عبارتند از:

- امکان اجرای کد یکسان روی CPU و GPU بدون نیاز به هیچ تغییری فراهم شده است.
- دارای API کاربرپسند است که ایجاد نمونه اولیه برای مدل‌های یادگیری عمیق را تسهیل می‌کند.
- دارای توابع از پیش تعریف شده‌ای برای شبکه‌های کانولوشنی (برای بینایی ماشین)، شبکه‌های برگشتی (برای پردازش دنباله) و هر ترکیبی از دو شبکه فوق است.
- از معماری‌های دلخواه شبکه پشتیبانی می‌کند: مدل‌های چند ورودی یا چند خروجی، به اشتراک‌گذاری لایه، به اشتراک‌گذاری مدل و غیره. در نتیجه، کراس برای ساخت هر نوع مدل یادگیری عمیق از شبکه GAN تا ماشین تورینگ عصبی مناسب است.

مقدمه‌ای بر کراس



برنامه‌نویسی در کراس: مرور سریع

روند کار کراس:

1) تعریف داده‌های آموزشی: تنسورهای ورودی و تنسورهای خروجی.
2) تعریف شبکه‌ای از لایه‌ها (همان تعریف مدل) که ورودی‌ها را به خروجی‌ها نگاشت می‌کند.
3) پیکربندی فرایند یادگیری با انتخاب تابع هزینه، یک بهینه‌ساز و تعدادی معیار برای ارزیابی
4) انجام عمل یادگیری روی داده‌های آموزشی با فراخوانی متد fit() از مدل

برنامه‌نویسی در کراس: مرور سریع

- مدل را می‌توان به دو روش تعریف کرد:
- **Sequential:** فقط مناسب دنباله پشت سرهمی از لایه‌ها که متداول‌ترین معماری شبکه است
- **API تابعی:** امکان ساخت معماری کاملاً دلخواه را فراهم می‌آورد

برنامه‌نویسی در کراس: مرور سریع

Sequential

```
from tensorflow.keras import models
from tensorflow.keras import layers
model = models.Sequential()
model.add(layers.Dense(32, activation='relu', input_shape=(784,)))
model.add(layers.Dense(10, activation='softmax'))
```

تابعی API

```
input_tensor = layers.Input(shape=(784,))
x = layers.Dense(32, activation='relu')(input_tensor)
output_tensor = layers.Dense(10, activation='softmax')(x)
model = models.Model(inputs=input_tensor, outputs=output_tensor)
```

برنامه‌نویسی در کراس: مرور سریع

```
from tensorflow.keras import optimizers
model.compile(optimizer=optimizers.RMSprop(lr=0.001),
                            loss='mse',
                            metrics=['accuracy'])
```

```
model.fit(input_tensor, target_tensor, batch_size=128, epochs=10)
```

چندین مثال کاربردی کراس

تخمین قیمت خانه با استفاده از داده‌های بنگاه‌های معاملاتی (رگرسیون)

دسته‌بندی موضوعی اخبار (دسته‌بندی چند کلاسی)

دسته‌بندی نظرات در مورد فیلم‌ها در دو سته نظرات منفی و نظرات مثبت (دسته‌بندی دودویی)

دسته‌بندی نظرات در مورد فیلم‌ها در دو سته نظرات منفی و نظرات مثبت

- دسته‌بندی نظرات کاربران در دو گروه نظر مثبت یا منفی
- دسته‌بندی دو کلاسی یا دودویی
- در این مثال با مجموعه داده IMDB کار خواهیم کرد
  - مجموعه‌ای از 50 هزار نظر منفی یا مثبت از پایگاه داده IMDB است.
  - این مجموعه به 25 هزار نظر برای آموزش و 25 هزار نظر برای آزمایش تقسیم شده است و هر مجموعه حاوی 50 درصد نظر مثبت و 50 درصد نظر منفی است.
- دلیل استفاده از مجموعه‌های جداگانه آموزش و آزمایش چیست؟

دسته‌بندی نظرات در مورد فیلم‌ها در دو سته نظرات منفی و نظرات مثبت

- ✓ هرگز نباید برای بررسی کارایی یک مدل یادگیری ماشین از داده‌هایی استفاده کنید که قبلاً توسط آن‌ها آموزش انجام شده است!

- ✓ عملکرد خوب یک مدل روی داده‌های آموزش‌دیده لزوماً بدین معنی نیست که روی داده‌هایی که هرگز ندیده است نیز همین عملکرد را خواهد داشت و از طرفی آنچه مهم است عملکرد خوب مدل روی داده‌های جدید است.

GOING TIME

3.5-classifying-movie-reviews

دسته‌بندی موضوعی اخبار (دسته‌بندی چند کلاسی)

- مجموعه‌ای از اخبار کوتاه و موضوع متناظر هر خبر که در سال 1986 توسط رویترز منتشر شده است.

- دسته‌بندی چندکلاسی

- 46 موضوع مختلف در این مجموعه داده موجود هستند

- برخی از آن‌ها بیشتر از بقیه تکرار شده‌اند، اما از هر موضوع حداقل 10 نمونه در مجموعه آموزشی وجود دارد.

GOING TIME

3.6-classifying-newswires

پیش‌بینی قیمت خانه‌ها: مثال رگرسیون

رگرسیون، نوع دیگری از مسائل یادگیری ماشین است که به جای پیش‌بینی یک برچسب گسسته، شامل پیش‌بینی یک مقدار پیوسته است.

- پیش‌بینی دمای هوای فردا، با داده‌های هواشناختی
- پیش‌بینی زمان تکمیل یک پروژه نرم‌افزار با استفاده از مشخصات آن.

با استفاده از داده‌های حومه شهر بوستون که متعلق به اواسط دهه 1970 است سعی خواهیم کرد قیمت متوسط مسکن را در این شهر پیش‌بینی کنیم. داده‌های این مجموعه، شامل ویژگی‌هایی همچون میزان جرم و جنایت، میزان مالیات محلی و غیره است.

پیش‌بینی قیمت خانه‌ها: مثال رگرسیون

- داده‌های مجموعه‌داده نسبتاً کم‌تر هستند: تنها شامل 506 داده است که به 404 نمونه آموزشی و 102 نمونه آزمایش تقسیم شده‌اند.

- هر ویژگی در نمونه‌های ورودی (به عنوان مثال، میزان جرم و جنایت) مقیاس متفاوتی دارد. به عنوان مثال، برخی از مقادیر به صورت درصد هستند و بین صفر و یک قرار می‌گیرند، برخی از مقادیر بین 1 تا 12، برخی بین صفر و 100 هستند.

GOING TIME

3.7-predicting-house-prices

تشکر

سوال؟

a.golzari@azaruniv.ac.ir

a.golzari@tabrizu.ac.ir

https://github.com/Amin-Golzari-Oskouei