دانشگاه صنعتی امیرکبیر
( پلی تکنیک تهران )

گزارش 8

# مقایسه روش‌های مختلف فراابتکاری برای بهینه سازی ضرایب یک شبکه عصبی برای حل مساله طبقه بندی Benchmark

مریم علیپور
۹۶۱۲۰۳۷

تصور کنید که شما یک محقق پزشکی هستید و برای مطالعه داده جمع آوری می کنید. شما اطلاعاتی درباره مجموعه ای از بیماران جمع آوری کرده اید که همه آن ها از یک بیماری خاص رنج می برند. در طی دوره درمان هر بیمار به یکی از ۵ دارو ، داروی A، B، C، X و Yواکنش مثبت میدهد و درمان میشود.

میخواهیم مدلی بسازیم که بفهمد کدام دارو برای بیمار جدید با همان بیماری مناسب است. مجموعه ویژگی های این مجموعه داده سن ، جنس ، فشار خون و کلسترول بیماران است و هدف پیدا کردن دارویی است که هر بیمار با آن درمان شود.

خواندن دیتا:

```
my_data = pd.read_csv("drug200.csv", delimiter=",")
my_data.head()
```

| | Age | Sex | BP | Cholesterol | Na_to_K | Drug |
|---|---|---|---|---|---|---|
| 0 | 23 | F | HIGH | HIGH | 25.355 | drugY |
| 1 | 47 | M | LOW | HIGH | 13.093 | drugC |
| 2 | 47 | M | LOW | HIGH | 10.114 | drugC |
| 3 | 28 | F | NORMAL | HIGH | 7.798 | drugX |
| 4 | 61 | F | LOW | HIGH | 18.043 | drugY |

حذف ستون target از ماتریس فیچرها و تبدیل ستون هایsex و bp (کتگوریکال) به وکتور one-hot:

```
X = my_data[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K']].values
X[0:5]
```

```
array([[23, 'F', 'HIGH', 'HIGH', 25.355],
       [47, 'M', 'LOW', 'HIGH', 13.093],
       [47, 'M', 'LOW', 'HIGH', 10.114],
       [28, 'F', 'NORMAL', 'HIGH', 7.798],
       [61, 'F', 'LOW', 'HIGH', 18.043]], dtype=object)
```

```
from sklearn import preprocessing
le_sex = preprocessing.LabelEncoder()
le_sex.fit(['F','M'])
X[:,1] = le_sex.transform(X[:,1])


le_BP = preprocessing.LabelEncoder()
le_BP.fit([ 'LOW', 'NORMAL', 'HIGH'])
X[:,2] = le_BP.transform(X[:,2])


le_Chol = preprocessing.LabelEncoder()
le_Chol.fit([ 'NORMAL', 'HIGH'])
X[:,3] = le_Chol.transform(X[:,3])

X[0:5]
```

```
array([[23, 0, 0, 0, 25.355],
       [47, 1, 1, 0, 13.093],
       [47, 1, 1, 0, 10.114],
       [28, 0, 2, 0, 7.798],
       [61, 0, 1, 0, 18.043]], dtype=object)
```

نگهداری ستون target در وکتور y:

```
y = my_data["Drug"]
y[0:5]
```

```
0    drugY
1    drugC
2    drugC
3    drugX
4    drugY
Name: Drug, dtype: object
```

برای یادگیری دیتا از مدل Multi-layer Perceptron در کتابخانه sklearn استفاده میکنیم.
آرگومان solver نوع اپتیمایزر شبکه را تعیین میکند که یکی از موارد زیر است:

۱. **lfgbs**: یک optimizer از خانواده‌ی quasi-Newton است. مخفف Limited-memory Broyden–Fletcher–Goldfarb–Shanno. این به روز رسانی ماتریس مشتق دوم را با ارزیابی شیب تقریبی می‌دهد. این فقط چند به روزرسانی آخر را ذخیره می‌کند، بنابراین حافظه را ذخیره می‌کند. با مجموعه داده‌های بزرگ خیلی سریع نیست. این از Scikit-learn نسخه ۰,۲۲,۰ به طور پیش فرض حل خواهد شد.

LBFGS Solver: optimizer in the family of quasi-Newton methods

Training ¶

```
MLP = MLPClassifier(solver='lbfgs')
MLP # it shows the default parameters
```

```
MLPClassifier(solver='lbfgs')
```

```
MLP.fit(X_trainset,y_trainset)
```

```
MLPClassifier(solver='lbfgs')
```

Predicting

```
predMlp = MLP.predict(X_testset)
```

```
print (predMlp [0:5])
print (y_testset [0:5])
```

```
['drugY' 'drugX' 'drugX' 'drugX' 'drugX']
40     drugY
51     drugX
139    drugX
197    drugX
170    drugX
Name: Drug, dtype: object
```

Evaluation

```
from sklearn import metrics
import matplotlib.pyplot as plt
print("Accuracy: ", metrics.accuracy_score(y_testset, predMlp))
print(metrics.classification_report(y_testset, predMlp))
```

```
Accuracy:  0.8833333333333333
              precision    recall  f1-score   support

       drugA       1.00      1.00      1.00         7
       drugB       0.62      1.00      0.77         5
       drugC       1.00      0.80      0.89         5
       drugX       0.95      0.90      0.93        21
       drugY       0.86      0.82      0.84        22

    accuracy                           0.88        60
   macro avg       0.89      0.90      0.88        60
weighted avg       0.90      0.88      0.89        60
```

۲. **sgd**: مخفف stochastic gradient descent است. یک روش ساده و در عین حال بسیار کارآمد برای فیت کردن طبقه‌بندی کننده‌های خطی و رگرسورها تحت توابع از دست دادن محدب مانند (خطی) ماشین‌های برداری و رگرسیون لجستیک است. حتی اگر SGD مدت زیادی در جامعه یادگیری ماشین وجود داشته باشد، اخیراً در زمینه یادگیری در مقیاس وسیع مورد توجه زیادی قرار گرفته است.SGD با موفقیت در زمینه مشکلات یادگیری ماشین در مقیاس بزرگ و پراکنده که معمولاً در طبقه بندی متن و پردازش زبان طبیعی وجود دارد، اعمال شد. با توجه به کم بودن داده‌ها ، طبقه‌بندی کنندگان در این ماژول به راحتی با بیش از ۱۰^۵ نمونه

آموزش و بیش از ۱۰^۵ ویژگی ، مقیاس بندی می‌کنند.

## SGD Solver: stochastic gradient descent

### Training

```
MLP = MLPClassifier(solver='sgd')
MLP # it shows the default parameters
```

```
MLPClassifier(solver='sgd')
```

```
MLP.fit(X_trainset,y_trainset)
```

```
MLPClassifier(solver='sgd')
```

### Predicting

```
predMlp = MLP.predict(X_testset)
```

```
print (predMlp [0:5])
print (y_testset [0:5])
```

```
['drugX' 'drugX' 'drugX' 'drugX' 'drugY']
40     drugY
51     drugX
139    drugX
197    drugX
170    drugX
Name: Drug, dtype: object
```

### Evaluation

```
from sklearn import metrics
import matplotlib.pyplot as plt
print("Accuracy: ", metrics.accuracy_score(y_testset, predMlp))
print(metrics.classification_report(y_testset, predMlp))
```

```
Accuracy:  0.6333333333333333
              precision    recall  f1-score   support

       drugA       0.00      0.00      0.00         7
       drugB       1.00      0.60      0.75         5
       drugC       0.00      0.00      0.00         5
       drugX       0.73      0.76      0.74        21
       drugY       0.54      0.86      0.67        22

    accuracy                           0.63        60
   macro avg       0.45      0.45      0.43        60
weighted avg       0.54      0.63      0.57        60
```

**adam:**

Adam یک الگوریتم بهینه سازی میزان یادگیری تطبیقی است که به طور خاص برای آموزش شبکه‌های عصبی عمیق طراحی شده است. آدام برای اولین بار در سال ۲۰۱۴ منتشر شد، در یک کنفرانس بسیار معتبر برای پزشکان یادگیری عمیق - ICLR 2015 ارائه شد. مقاله حاوی نمودارهای بسیار امیدوار کننده‌ای بود که نشان دهنده پیشرفت‌های چشمگیر عملکرد از نظر سرعت آموزش است. با این حال، پس از مدتی مردم شروع به مشاهده کردند که در بعضی موارد آدام واقعاً راه حل بدتری نسبت به نزول شیب تصادفی پیدا می‌کند. تحقیقات زیادی برای رفع مشکلات آدم انجام شده است. این الگوریتم‌ها از قدرت روش‌های نرخ یادگیری انطباقی برای یافتن نرخ یادگیری فردی برای

هر پارامتر استفاده می‌کنند. این همچنین دارای مزایای Adagrad است، که در تنظیمات با شیب کم بسیار خوب عمل می‌کند، اما در بهینه‌سازی غیر محدب شبکه‌های عصبی و RMSpropتلاش می‌کند، که برای حل برخی از مشکلات Adagrad حل می‌شود و واقعاً کار می‌کند خوب در تنظیمات آنلاین. با توجه به مقاله A Peek at Trends in Machine Learning از آندره کارپاتی، adamاز محبوبیت بالایی برخوردار شده است.

Adam Solver: stochastic gradient-based optimizer proposed by Kingma, Diederik, and Jimmy Ba

Training

```
MLP = MLPClassifier(solver='adam')
MLP # it shows the default parameters
```

```
MLPClassifier()
```

```
MLP.fit(X_trainset,y_trainset)
```

```
MLPClassifier()
```

Predicting

```
predMlp = MLP.predict(X_testset)
```

```
print (predMlp [0:5])
print (y_testset [0:5])
```

```
['drugY' 'drugX' 'drugX' 'drugX' 'drugY']
40     drugY
51     drugX
139    drugX
197    drugX
170    drugX
Name: Drug, dtype: object
```

Evaluation

```
from sklearn import metrics
import matplotlib.pyplot as plt
print("Accuracy: ", metrics.accuracy_score(y_testset, predMlp))
print(metrics.classification_report(y_testset, predMlp))
```

```
Accuracy:  0.6666666666666666
              precision    recall  f1-score   support

       drugA       0.00      0.00      0.00         7
       drugB       0.67      0.80      0.73         5
       drugC       0.00      0.00      0.00         5
       drugX       0.82      0.86      0.84        21
       drugY       0.58      0.82      0.68        22

    accuracy                           0.67        60
   macro avg       0.41      0.50      0.45        60
weighted avg       0.55      0.67      0.60        60
```

همانطور که مشاهده میکنیم دقت مدل با lfgbs ۸۸٪ شد و تقریبا نسبت به دوحالت دیگر ۲۰ درصد بهتر عمل کرده.

منابع:

**Dataset:** https://www.kaggle.com/ibrahimbahbah/drug200