

پروژه پایانی

مریم عظیم پور

۹۵۵۲۱۳۳۳

در این مسئله از روش MLP استفاده شده است. شبکه‌ی RBF نیز برای این مسئله مناسب است.

ابتدا ورودی‌ها از فایل اکسل خوانده شده و بعد از نرمالسازی و حذف داده‌ی مربوط به شماره‌ی دانشجویی افراد که تاثیری در معدل و خروجی شبکه ندارد، به دو بخش داده‌ی تست و داده‌ی train تقسیم میشود و سپس شبکه با دیتای train آموزش داده میشود و بعد با دیتای تست، ارزیابی میشود.

```
1 data = pd.read_excel (r'Elearning-Data-cut.xls',sheet_name='input')
2 output = pd.read_excel (r'Elearning-Data-cut.xls',sheet_name='out_True value ')
3 output = output.values
4 binaryOutput = pd.read_excel (r'Elearning-Data-cut.xls',sheet_name='out_Binary value _drop ')
5 binaryOutput = binaryOutput.values.T
6 binaryOutput = binaryOutput[0]
7 data = data.drop(['STUDENTN'], axis=1)
8 data = normalization(data)
9
10 x_train = data[:600]
11 x_test = data[600:]
12
13 y_train = output[:600]
14 y_test = output[600:]
15
16 binaryY_train = binaryOutput[:600]
17 binaryY_test = binaryOutput[600:]
```

نرمالسازی داده‌ها با فراخوانی تابع normalization() انجام میشود. این تابع داده‌های باینری مانند ساکن تهران بودن یا نبودن را به 1 و -1 تبدیل میکند و داده‌های عددی را با استفاده از تابع آماده‌ی MinMaxScaler() نرمالسازی میکند. همچنین داده‌های null را به 0 تبدیل میکند.

```
1 def normalization(data):
2     data[np.isnan(data)] = 0
3     x = data.values
4     min_max_scaler = preprocessing.MinMaxScaler().fit_transform(x)
5     data = pd.DataFrame(min_max_scaler,columns=data.columns)
6     binaryColumns =['GENDER','MARRY','SHAGHEL','TEHRAN','URBEN','SHARI','HAZ_ME','HAZ_FAM','HAZI_MF','SPEEDAB','LAPTOP','MOBILE']
7     for i in binaryColumns:
8         data[i] = [1 if j==1 else -1 for j in data[i]]
9     return data.values
```

مدل تخمین معدل

شبکه عصبی استفاده شده برای پیش‌بینی معدل، یک شبکه‌ی MLP ۳ لایه است که در لایه‌های اول و دوم، تعداد نورون‌ها به اندازه‌ی ابعاد ورودی است و لایه‌ی آخر یک نورون دارد که خروجی آن عددی بین ۰ تا ۲۰ به عنوان معدل پیش‌بینی شده است. همچنین در این شبکه از dropout استفاده شده تا از overfit تا حدودی جلوگیری شود.

در لایه‌های اول و دوم از توابع فعال‌سازی (activation function) relu استفاده شده است و لایه‌ی آخر که لایه‌ی خروجی است تابع فعال‌سازی ندارد.

```
1 #model
2 model = Sequential()
3
4 model.add(Dense(units=data.shape[1], activation='relu'))
5 model.add(Dropout(0.2))
6 model.add(Dense(units=data.shape[1], activation='relu'))
7 model.add(Dropout(0.1))
8 model.add(Dense(units=1))
```

در این شبکه از اپتیمایزر adam و تابع خطای mean_squared_error استفاده شده و فرآیند train در ۲۰۰۰ اپاک و با batch_size = 200 انجام شده است.

عدد مناسب برای اپاک و batch_size و همچنین تعداد نورون‌های مناسب برای لایه‌ها با آزمون و خطا به دست آمده است و از میان چندین مدل با تعداد نورون‌ها و پارامترهای مختلف، این مدل بیشترین دقت را داشته است. این شبکه می‌تواند با میانگین خطای 1 تا 1.5 نمره، معدل را پیش‌بینی کند.

خروجی

برای اجرای برنامه، اطلاعات مربوط به دانشجو در قالب فایل اکسل به برنامه داده می‌شود و بعد از نرمال‌سازی داده‌ها، با استفاده از مدل طراحی شده معدل دانشجو در ترم بعد پیش‌بینی می‌شود. به دلیل خطای مدل، ممکن است معدل دانشجو عددی بیشتر از ۲۰ پیش‌بینی شود که در این صورت معدل ۲۰ در نظر گرفته می‌شود. خروجی نهایی به شکل زیر است:

_____accuracy_____

average error: 1.09 point

_____results_____

	student number	GPA
0	88222302	18.939693
1	87261187	12.251963
2	90100106	11.779272
3	90291063	20.000000
4	88271108	12.648962