به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی برق

هوش مصنوعی و محاسبات زیستی ـ دکتر حاجی پور

نیم‌سال اول ۱۳۹۹

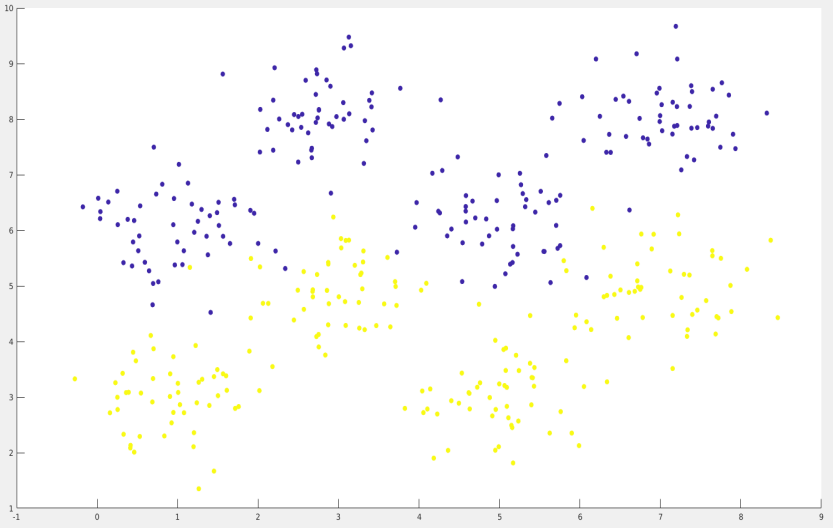
گزارش بخش کامپیوتری تمرین سری دوم

امید شرفی (۹۶۱۰۱۸۳۸)

فهرست مطالب

۱ Learning using RBF and MLP models

۱.۱ نمایش داده ها



شکل ۱: نمایش داده ها

۲.۱ آماده سازی داده ها

برای این بخش چون گفته نشده بود که داده ها به صورت رندم در آرایه چیده شدن، با اعمال یک پرمیوتیشن رندوم و متناسب با درصت تقسیم بندی گفته شده در سوال داده ها و لیبل ها متناظر را برای اعتبارسنجی و آموزش جدا میکنیم.

البته میتوانستیم همین کار را در شبکه هایی که در ادامه طراحی میکنیم هنگام فرآیند آموزش اش به صورت پارامتر در توابع آماده متلب وارد کنیم ولیکن چون در صورت تمرین گفته شده و همچنین از این نظر که دسته بندی ما برای بخش های مختلف ثابت میماند ما این تقسیم بندی را همینجا انجام دادیم.

۳.۱ شبکه MLP تک لایه

برای پیدا کردن بهترین پارامتر تعداد نورون های لایه ی پنهان شبکه را داخل یک حلقه برای اعداد مختلف تعداد نورون ها در لایه ی پنهان تست میکنیم.

```
% we will search in for loop to find best parameters
Neurons = [1:30 50 100 200 500];

error_min = 10000;
best_param = [];

for neuron = Neurons
    net = feedforwardnet(neuron);

    % Setup Division of Data for Training, Validation, Testing. We have indexes
    % of each part from the last part so we use divideind as divideFcn
    net.divideFcn = 'divideind';
    net.divideParam.trainInd = index_train;
    net.divideParam.valInd = index_test;
    net.divideParam.testInd = [];

    % Train the Network
    [net,tr] = train(net, TrainingData, TrainingLabels);

    % Find best parameters for model
    Y_o = net(X_test);
    if(norm(Y_o - Y_test) < error_min)
        error_min = norm(Y_o - Y_test);
        best_param = neuron;
    end

end
```

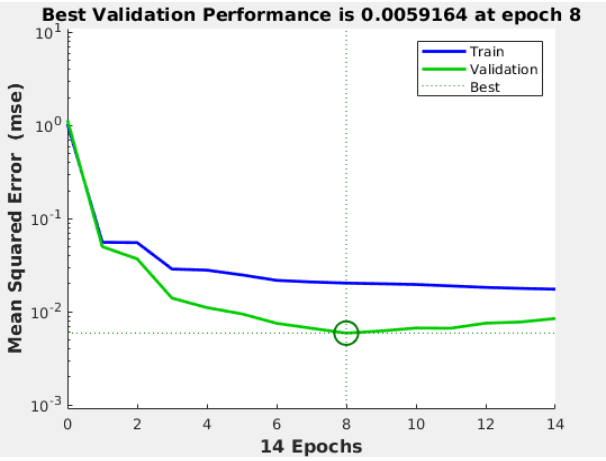
شکل ۲: جست و جو برای یافتن هایپرپارامترهای بهینه ی مدل

متناسب با خطای داده های ارزیابی بهترین هایپرپارامتر مدل را پیدا میکنیم.

```
Best hyper parameters of the MLP model is :
 Number of hidden neurons = 21
```

شکل ۳: بهترین هایپارامترهای مدل

و نتایج عملکرد شبکه ی مطلوب به صورت زیر به دست می آید.



شکل ۴: عملکرد شبکه MLP با بیست و یک نورون لایه ی پنهان

۴.۱ شبکه RBF تک لایه

برای این که بتوانیم بهترین هایپرپارامترهای شبکه ی RBF نرمال را به دست بیاوریم داخل دو حلقه به جستجو خواهیم پرداخت. توجه شود که برای آنکه MN ما را محدود کند ما هدف خطا را صفر قرار میدهیم. در عین حال مبنای الگوریتم بر اساس معادلات به دست آمده برای ورودی ها هست و در نتیجه قرار دادن MN بالای ۳۲۰ که اندازه ورودی هست بی معنی بوده چون عملا برای این حالت خطا روی داده ها توسط RBF صفر میشود.

در ادامه با استفاده از تابع sim خطا را روی داده هایی که به عنوان ارزیابی جدا کردیم آزمایش کرده و مدل بهینه را بر این اساس انتخاب میکنیم.

```
% we will search in for loop to find best parameters
Spreads = [0.001, 0.01, 0.1, 1, 10];
MNs = [1:30, 50, 80, 100, 200, 320];

error_min = 10000;
best_param = [];

for spread = Spreads
    for MN = MNs
        net = newrb(X_train, Y_train, 0, spread, MN);
        Y_o = sim(net, X_test);
        if(norm(Y_o - Y_test) < error_min)
            error_min = norm(Y_o - Y_test);
            best_param = [spread MN];
        end
    end
end
```

شکل ۵: جست و جو برای یافتن هایپرپارامترهای بهینه ی مدل

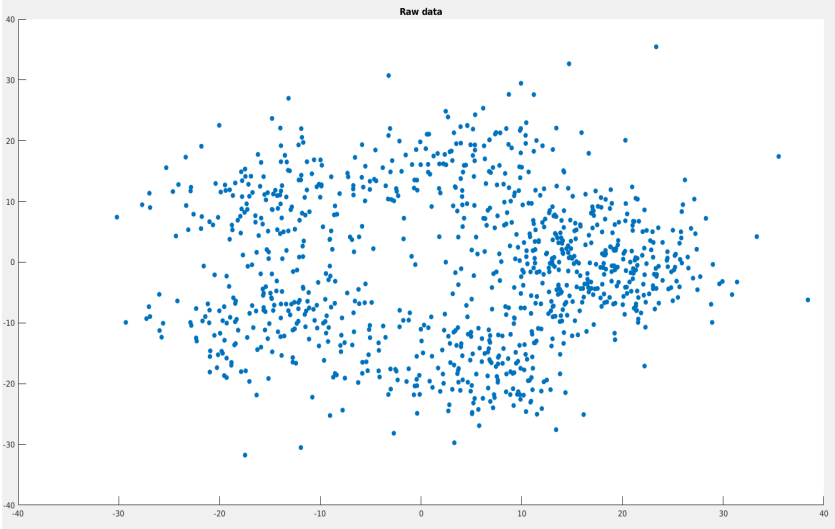
متناسب با خطای داده های ارزیابی بهترین هایپرپارامتر مدل را پیدا میکنیم و و نتایج عملکرد شبکه ی مطلوب به صورت زیر به دست می آید.

```
Best hyper parameters of the RBF model is :
 Spread = 1
 MN = 25
The best RBF model norm2 error : 1.401961e+00
```

شکل ۶: بهترین هایپارامترهای مدل و نتیجه ی آن

۲ Clustering using kMeans and LVQ algorithms

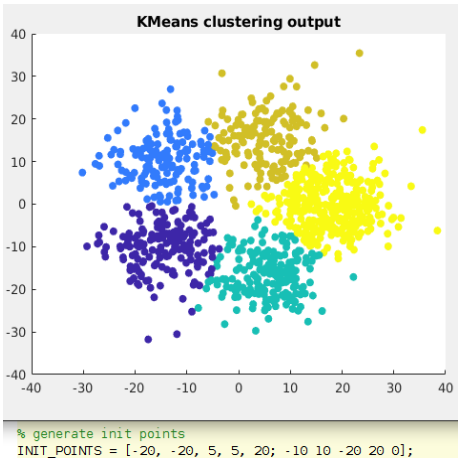
۱.۲ بررسی فرم پخش شدگی داده ها در صفحه



شکل ۷: نمایش داده ها

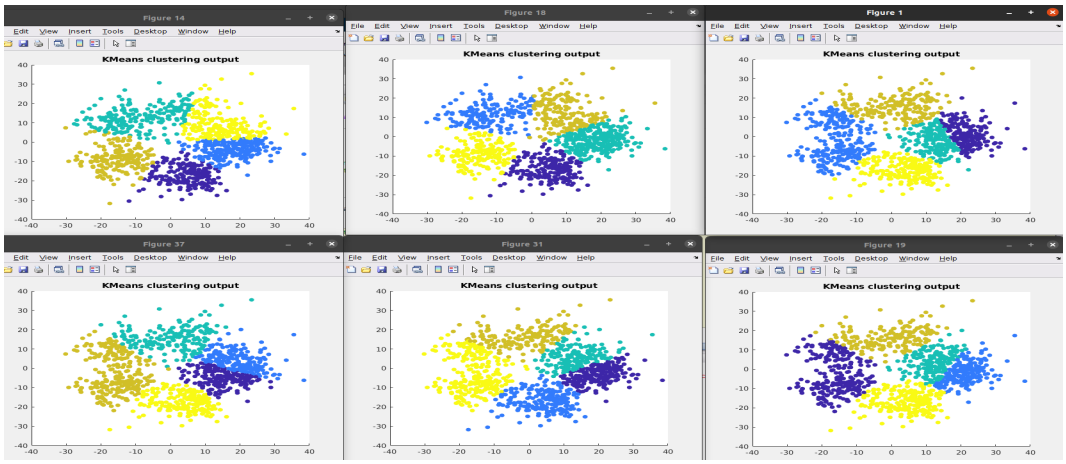
۲.۲ kMeans

پیاده سازی الگوریتم که در تابع مربوطه موجود هست. برای فاصله نرم دو محاسبه میشود و سپس برای مختصات هر مرکز میانگین داده های اختصاص شده به آن مرکز را قرار میدهیم. خروجی با ۱۰ تکرار و مراکز مشخص شده در شکل زیر آورده شده است.



شکل ۸: خروجی دسته بندی با روش kMeans

همچنین برای آنکه تاثیر نقاط اولیه را ببنیم با حفظ تعداد تکرارهای الگوریتم و تعداد دسته ها، به صورت رندوم از داده های ورودی به تعداد دسته ورودی برداشته نقاط اولیه را انتخاب کردیم. اینمار را پنجاه دفعه تکرار کردیم که نمونه ای از دسته بندی هایی که شکلشون تا حدودی با دسته بندی بتدایی ما متفاوت بودن رو در شکل زیر آورده ایم :

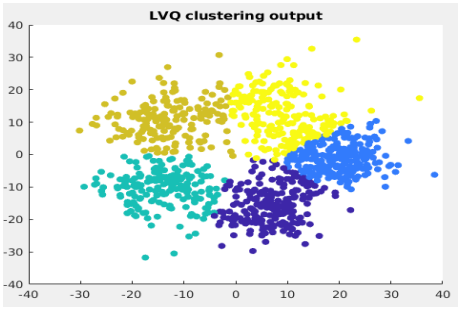


شکل ۹: تعدادی از دسته بندی های متفاوت kMeans

در نتیجه نقاط اولیه ی الگوریتم و تعداد دفعات تکرار مهم خواهند بود. همانطور که مشاهده میشود مثلا در شکل های بالا به نظر میرسد که تراکم نقاط اولیه در سمت راست بیشتر بوده و در نتیجه با توجه به آن که دو دسته ی سمت چپ خیلی هم مجزا نبوده اند فرصت کردند تا یک کلاستر بزرگ شکل دهند و در نتیجه یکی از کلاستر های سمت راست به اجبار با رقابت بین دو مرکز به دو دسته تقسیم شده است.

LVQ ۳.۲

خروجی الگوریتم LVQ با ۳۰ تکرار و ۵ دسته به شکل زیر درآمد که همانطور که مشاهده میشود دسته بندی با دقت خیلی خوبی صورت گرفته است.



شکل ۱۰: خروجی دسته بندی با روش LVQ