



دانشگاه تهران

دانشکده فنی-مهندسی کامپیوتر

دپارتمان الگوریتم ها و محاسبات

گزارش تمرین شماره ی سه

طراحی الگوریتم انتشار بازگشتی

نیلوفر آقایی ابیانه

۸۱۰۸۹۰۰۰۱

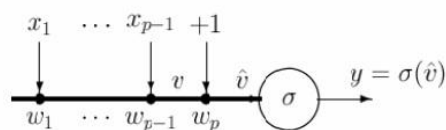
چکیده

در این پروژه، از روش های یادگیری^۱ برای طبقه بندی مجموعه ای از داده ها استفاده شده و نتایج حاصل تحلیل می شوند. در اینجا سه الگوریتم گیت OR، گیت XOR، و الگوریتم انتشار بازگشتی شبیه سازی شده اند. برای این کار از روش یادگیری پرسپترون چند لایه^۲ استفاده شده است. در الگوریتم های گیت OR، گیت XOR از تابع فعال سازی^۳ پله واحد و در الگوریتم انتشار بازگشتی^۴ از تابع فعال سازی sigmoid استفاده شده است.

۱. مقدمه

در علم کامپیوتر، الگوریتم های مختلفی برای طبقه بندی وجود دارد، یکی از این الگوریتم ها روش های یادگیری است. در این روش با استفاده از مجموعه ای داده که به آن داده ی آموزش^۵ گفته می شود، ماشین، آموزش داده می شود؛ سپس با کمک ماشین به دست آمده هر نمونه تست بررسی شده و طبقه بندی می شود.

در روش یادگیری ابتدا بردار ورودی که مجموعه ویژگی ها ی یک نمونه آموزش است، در مجموعه ای از وزن ها ضرب می شود. سپس حاصل ضرب ها با هم جمع شده و به عنوان ورودی تابع فعال سازی داده می شوند (شکل ۱). در آخر یک خروجی به دست می آید که با خروجی مطلوب (آخرین ویژگی نمونه آموزش که در طبقه بندی معادل با کلاس نمونه مورد نظر است) ممکن است اختلاف (خطا) داشته باشد. در این صورت این اختلاف برای به روز کردن وزن ها به کار می رود؛ این روند برای نمونه آموزش بعدی مجدداً تکرار می شود. این مشی تا زمانی که ماشین خطا نداشته یا خطا به یک حد ثابتی برسد، ادامه می یابد.



شکل ۱- ضرب نقطه ای بردار ورودی در بردار وزن ها

که $x = [x_1, x_2, x_3, \dots, x_{p-1}]$ بردار ورودی و $W = [w_1, w_2, w_3, \dots, w_p]$ بردار وزن ها می باشند.

^۱ Learning

^۲ Multi Level Perceptron

^۳ Activation Function

^۴ Back- propagation

^۵ Training Data

۲. شبیه سازی گیت OR

در اینجا نشان داده می شود که چگونه می توان از گیت OR ، برای طبقه بندی استفاده کرد. تابع فعال سازی در این شبیه سازی تابع پله واحد است و خطا از فرمول زیر محاسبه می شود؛

$$w(n + 1) = w(n) + \eta \cdot e(n) \cdot x(n)$$

که در آن η ، نرخ آموزش^۱، $w(n)$ بردار وزن ها ، $e(n)$ خطا، و $x(n)$ بردار ورودی است.

مجموعه آموزش ، نرخ آموزش ، وزن های اولیه به صورت زیر است :

$$\text{Trainingset} = [0 \ 0 \ 0 , 0 \ 1 \ 1 , 1 \ 0 \ 1 , 1 \ 1 \ 1]$$

$$\text{LR} = 0.2$$

$$W = [0.7 , -0.3 , 0.5]$$

که این ماشین ۱۵ بار اجرا می شود و پس از این تعداد تکرار ماشین آموزش دیده و خطای آن صفر می شود. وزن های بهینه به دست آمده در جدول ۱ آمده است.

وزن های بهینه	۰.۷۰۰۰	۰.۱۰۰۰	-۰.۱۰۰۰
---------------	--------	--------	---------

جدول ۱- وزن های بهینه برای OR_GATE با $\text{LR}=0.2$ و وزن های اولیه ۰.۷، -۰.۳، ۰.۵ و

در واقع در صورتی که بردار وزن اولیه به صورت $W=[0.7 , 0.1 , -0.1]$ باشد، با ورودی های $\text{trainingset}=[0 \ 0 \ 0 , 0 \ 1 \ 1 , 1 \ 0 \ 1 , 1 \ 1 \ 1]$ خروجی $[0 , 1 , 1 , 1]$ حاصل می شود.

۳. شبیه سازی گیت XOR

در اینجا نشان داده می شود که چگونه نمی توان از گیت XOR ، برای طبقه بندی استفاده کرد.

تابع فعال سازی در این شبیه سازی تابع پله واحد است و خطا از فرمول زیر محاسبه می شود؛

^۱ Learning Rate

$$w(n+1) = w(n) + \eta \cdot e(n) \cdot x(n)$$

که η ، نرخ آموزش، $w(n)$ بردار وزن ها، $e(n)$ خطا، و $x(n)$ بردار ورودی است.

مجموعه آموزش، نرخ آموزش وزن های اولیه به صورت زیر است :

$$\text{Trainingset} = [0 \ 0 \ 0 ; 0 \ 1 \ 1 ; 1 \ 0 \ 1 ; 1 \ 1 \ 0]$$

$$\text{LR} = 0.2$$

$$W = [0.7 ; -0.3 ; 0.5]$$

که این ماشین ۱۵ بار اجرا می شود و پس از این تعداد تکرار ماشین آموزش دیده اما خطای آن صفر نمی شود. وزن های به دست آمده در جدول ۲ آمده است.

وزن های به دست آمده	-۰.۱۰۰۰	-۰.۳۰۰۰	۰.۱۰۰۰
---------------------	---------	---------	--------

جدول ۲- وزن های به دست آمده برای XOR_GATE با $\text{LR}=0.2$ و وزن های اولیه ۰.۷، -۰.۳ و ۰.۵

در واقع در صورتی که بردار وزن اولیه به صورت $W = [-0.1, -0.3, 0.1]$ باشد، با ورودی های $\text{trainingset} = [0 \ 0, 0 \ 1, 1 \ 0, 1 \ 1]$ خروجی $[1, 1, 1, 1]$ حاصل می شود.

در واقع از پرسپترون تنها برای طبقه بندی های خطی می توان استفاده کرد؛ و در شرایط غیر خطی می توان از پرسپترون چند لایه ای استفاده کرد.

۴. پیدای سازی الگوریتم انتشار بازگشتی با استفاده از پرسپترون چند لایه ای

در اینجا الگوریتم انتشار بازگشتی با استفاده از پرسپترون چند لایه ای و تابع فعال سازی sigmoid پیاده سازی می شود. این الگوریتم طبقه بندی را روی مجموعه داده ای^۱ iris و satimage انجام می دهد؛ که iris مجموعه داده ای با 5×150 بوده و satimage مجموعه داده

^۱ dataset

ای با ۶۴۳۵×۳۷ است. پس در iris و satimage بردارهای ورودی به ترتیب به صورت $x = [x_1, x_2, x_3, x_4]$ و $x = [x_1, x_2, \dots, x_{36}]$ می باشند.

در اینجا از یک پرسپترون سه لایه ای استفاده می شود که تعداد عناصر لایه نخست، n_1 ، برابر با تعداد عناصر ویژگی، تعداد عناصر لایه دوم، n_2 ، مشخص نبوده، و تعداد عناصر لایه سوم، n_3 ، برابر با تعداد کلاس ها می باشد.

در محیط MATLAB با استفاده از تابع n2finding مقدار مناسب n_2 ، به دست می آید. این تابع ابتدا مقدار n_2 ، را برابر ۲، قرار می دهد. در ادامه تا دو برابر تعداد ویژگی ها در هر مرحله مقدار n_2 یک واحد افزایش داده و به ازای آن n_2 را آموزش می دهد. در آخر از بین n_2 های به دست آمده مقداری که بیشترین کارایی را دارد به عنوان بهترین n_2 در نظر می گیرد و باقی الگوریتم را با آن n_2 اجرا می کند. البته مقدار n_2 به نرخ آموزش نیز بستگی دارد.

پس از اینکه مقدار n_2 مشخص شد، الگوریتم ۲۵٪ کل داده را به عنوان تست در نظر گرفته و باقی داده را به چهار قسمت مساوی تقسیم کرده و در چهار مرحله، هر بر یک قسمت را به عنوان داده ی اعتبار سنجی^۱ و سه قسمت دیگر را به عنوان داده های آموزش برای آموزش ماشین استفاده می کند؛ که به این روش Multifold cross-validation گفته می شود

الگوریتم، با استفاده از داده های آموزش ماشین و تابع MLP ماشین را آموزش می دهد؛ سپس از ۲۵٪ داده ی باقی مانده برای تست ماشین استفاده می کند و این روند تا زمانی که اختلاف خطا دو نسل^۲ متوالی کمتر از ۰.۰۰۰۱ نشود ادامه پیدا می کند؛ در نهایت کارایی ماشین با استفاده از تابع evaluation محاسبه می شود.

این الگوریتم برای به روز کردن وزن ها از فرمول زیر استفاده می کند

$$w(n+1) = w(n) + \eta \cdot e \cdot f(1-f)$$

که $f(1-f)$ مشتق تابع sigmoid، η ، نرخ آموزش، $w(n)$ بردار وزن ها و e خطا است.

^۱ Validation data

^۲ epoch

نکته ای که باید به آن توجه کرد این است که الگوریتم در ابتدا وزن ها به صورت تصادفی در بازه ی ۰-۱ انتخاب می کند و داده های هر مجموعه داده ای را نرمال سازی می کند.

برای اجرای این الگوریتم در محیط MATLAB از توابع زیر استفاده شده است:

- MAIN : تابع اصلی؛ نوع مجموعه داده ای و مقدار نرخ آموزش را از کاربر می گیرد
- N2finding : محاسبه ی مقدار n_2 مناسب؛ از مقدار n_2 ، برابر ۲، شروع کرده در ادامه تا دو برابر تعداد ویژگی ها، در هر مرحله مقدار n_2 یک واحد افزایش داده و به ازای آن n_2 را آموزش می دهد. در آخر از بین n_2 های به دست آمده مقداری که بیشترین کارایی را دارد به عنوان بهترین n_2 در نظر می گیرد .
- Insertionsort_onlabels : مرتب سازی آرایه ورودی بر اساس کلاس ها
- Getclassesandfeatures : تعیین مقدار n_1 و n_3
- Initiaweight : مقدار دهی اولیه وزن ها
- MLP : تابع اصلی برای یادگیری ماشین به ازای هر نمونه train؛
- Normalize : نرمال سازی عناصر یک مجموعه داده
- Evaluation : تابع بررسی خروجی ماشین (کارایی)
- Selectingtestandtrain : تعیین مجموعه داده ای های train و تست
- Earlystoppingmethod : داده ها به دو دسته ی آموزش و اعتبار سنجی تقسیم می کند و پس از آموزش ماشین با استفاده از داده های آموزش ، از داده های اعتبار سنجی برای ارزیابی کارایی ماشین استفاده می کند.

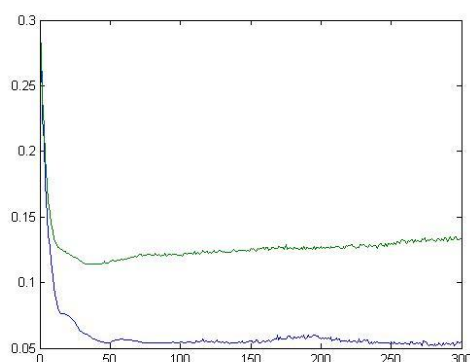
۵. آزمایش ها

با اجرای تابع MAIN روی مجموعه داده ای satimage با نرخ آموزش ۰.۲ و n_2 مناسب جدول ۳ حاصل می شود. ابتدا پس از جدا کردن داده های تست از آموزش، داده های باقی مانده برای آموزش به چهار قسمت تقسیم می شوند و هربار سه قسمت برای آموزش و قسمت دیگر برای داده ی اعتبار سنجی به کار گرفته می شوند.

نسل	۱۵۷	۱۰۵	۱۶۷	۲۵
کارایی	۷۸.۶۰۷۰	۷۰.۷۵۳۹	۶۴.۹۲۵۴	۶۶.۶۹۴۳

جدول ۳- اجرای الگوریتم MLP روی مجموعه داده ای satimage و $LR = 0.2$ با انتخاب n_2 برابر با ۵ به عنوان n_2 مناسب

با توجه به جدول ۳ میانگین و واریانس به ترتیب برابر است با، ۷۰.۲۴۵۱ و ۳۷.۰۲۹۲. شکل ۲ نمودار حاصل از تابع **earlystoppingmethod** را نشان می دهد. شکل ۳ نمودار های حاصل از اجرای چهار بار تابع MLP است.



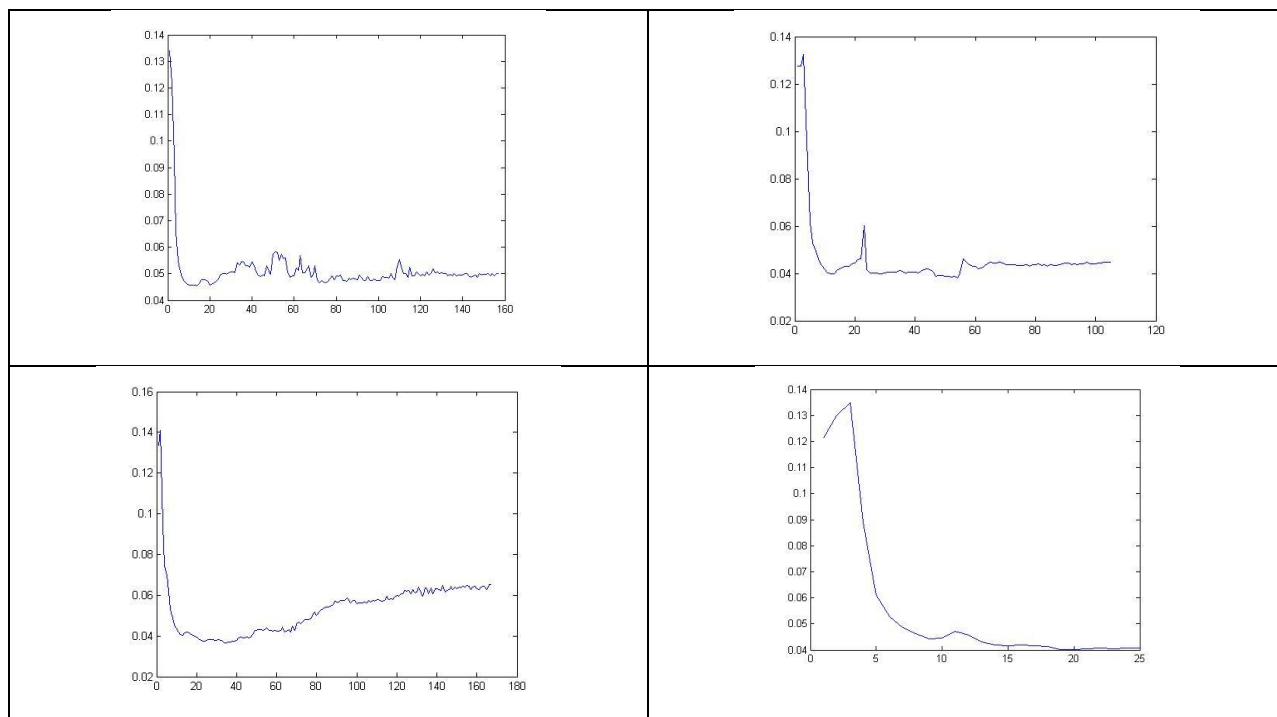
شکل ۲- نمودار حاصل از تابع **earlystoppingmethod** با $LR=0.2$ و $n_2=5$ به عنوان n_2 مناسب

با اجرای تابع MAIN روی مجموعه داده ای satimage با نرخ آموزش ۰.۳ و n_2 مناسب جدول ۴ حاصل می شود. ابتدا پس از جدا کردن داده های تست از آموزش، داده های باقی مانده برای آموزش به چهار قسمت تقسیم می شوند و هربار سه قسمت برای آموزش و قسمت دیگر برای داده ی اعتبار سنجی به کار گرفته می شوند.

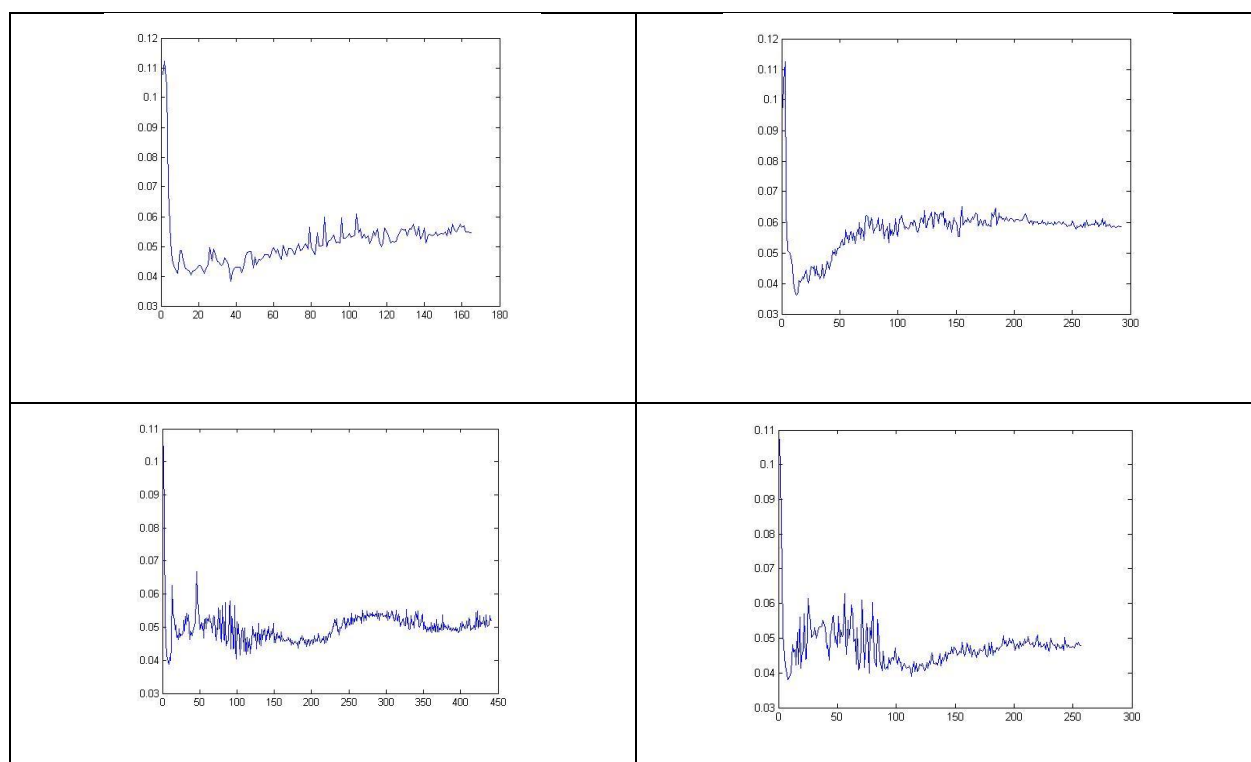
نسل	۱۶۵	۲۹۲	۴۴۱	۲۵۷
کارایی	۷۴.۸۷۵۶	۷۷.۷۹۶۲	۷۳.۵۴۸۹	۷۷.۲۱۶۲

جدول ۴- اجرای الگوریتم MLP روی مجموعه داده ای satimage و $LR = 0.3$ با انتخاب n_2 برابر با ۴ به عنوان n_2 مناسب

با توجه به جدول ۴ میانگین و واریانس به ترتیب برابر است با، ۷۵.۸۵۹۲ و ۳.۹۶۶۱. شکل ۴ نمودار های حاصل از اجرای چهار بار تابع MLP است.



شکل ۳- خطا های حاصل از چهار بار اجرا ی اجرای الگوریتم MLP روی مجموعه داده ای satimage و $LR = 0.2$ با انتخاب n_r برابر با ۵ به عنوان n_r مناسب



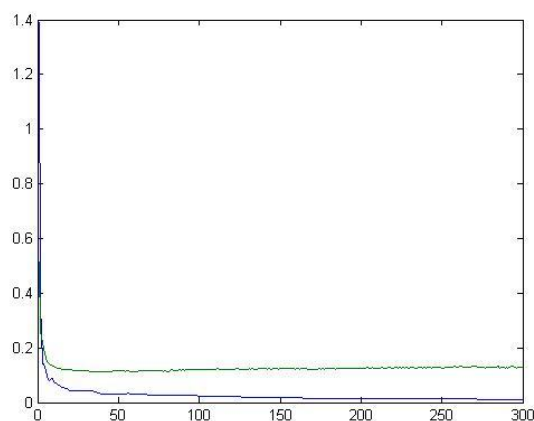
شکل ۴- خطا های حاصل از چهار بار اجرا ی اجرای الگوریتم MLP روی مجموعه داده ای satimage و $LR = 0.3$ با انتخاب n_r برابر با ۴ به عنوان n_r مناسب

با اجرای تابع MAIN روی مجموعه داده ای satimage با نرخ آموزش ۰.۴ و n_2 مناسب جدول ۵ حاصل می شود. ابتدا پس از جدا کردن داده های تست از آموزش، داده های باقی مانده برای آموزش به چهار قسمت تقسیم می شوند و هربار سه قسمت برای آموزش و قسمت دیگر برای داده ی اعتبار سنجی به کار گرفته می شوند.

نسل	۳۷	۲۸	۱۴	۷۸
کارایی	۴۴.۰۲۹۹	۴۳.۵۷۹۱	۴۵.۰۲۴۹	۵۱.۵۵۱۲

جدول ۵- اجرای الگوریتم MLP روی مجموعه داده ای satimage و $LR = ۰.۴$ با انتخاب n_2 برابر با ۴۲ به عنوان n_2 مناسب

با توجه به جدول ۵ میانگین و واریانس به ترتیب برابر است با، ۵۱.۵۵۱۲ و ۲۱۵.۸۶۰۷. شکل ۵ نمودار حاصل از تابع earllystoppingmethod را نشان می دهد. شکل ۶ نمودارهای حاصل از اجرای چهار بار تابع MLP است.

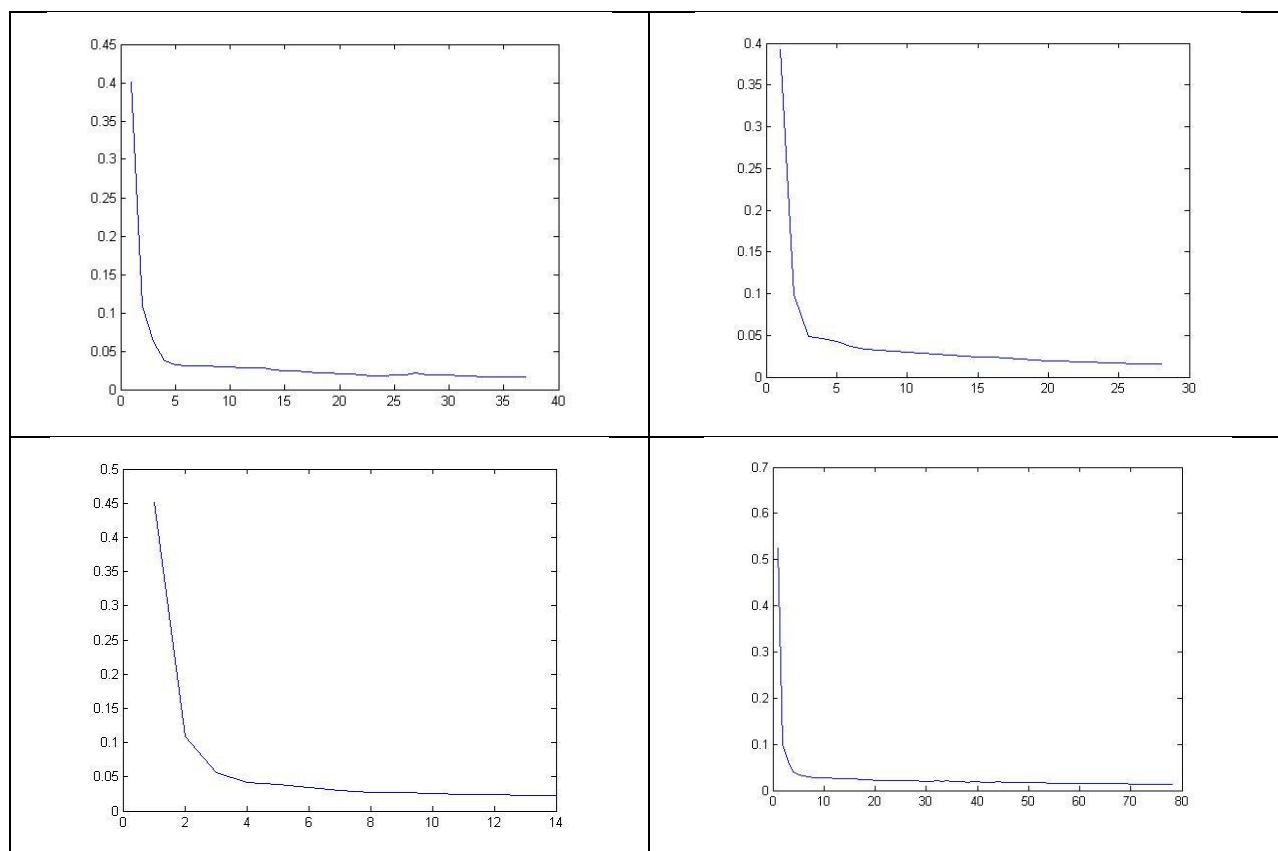


شکل ۵- نمودار حاصل از تابع earllystoppingmethod با $LR=۰.۴$ و $n_2=۴۲$ به عنوان n_2 مناسب

با اجرای تابع MAIN روی مجموعه داده ای iris با نرخ آموزش ۰.۲ نتایج در جدول ۶ حاصل شد. ابتدا پس از جدا کردن داده های تست از آموزش، داده های باقی مانده برای آموزش به چهار قسمت تقسیم می شوند و هربار سه قسمت برای آموزش و قسمت دیگر برای داده ی اعتبار سنجی به کار گرفته می شوند.

نسل	۵۴۹	۵۴۴	۳۸۹	۳۱۵
کارایی	۱۰۰	۱۰۰	۹۲.۸۵۷۱	۹۶.۴۲۸۶

جدول ۶- اجرای الگوریتم MLP روی مجموعه داده ای iris و $LR = ۰.۲$ با انتخاب n_2 برابر با ۳ به عنوان n_2 مناسب.



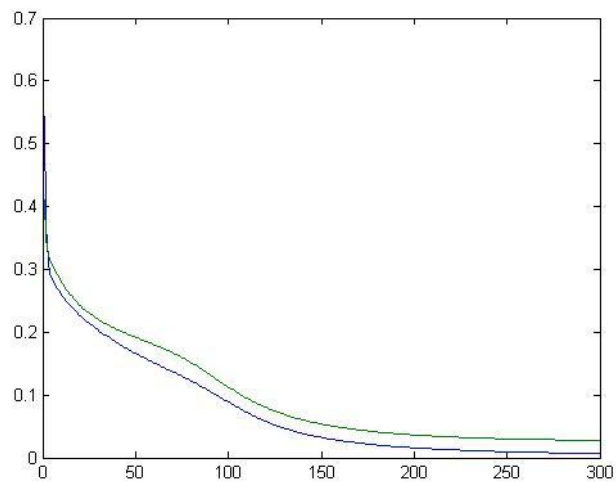
شکل ۶- خطا های حاصل از چهار بار اجرای الگوریتم MLP روی مجموعه داده ای satimage و $LR = 0.3$ با انتخاب n_2 برابر با ۴۲ به عنوان n_2 مناسب

با توجه به جدول ۶ میانگین و واریانس به ترتیب برابر است با، ۹۷.۳۲۱۴ و ۱۱.۶۹۲۲. شکل ۷ نمودار حاصل از تابع `earlystoppingmethod` را نشان می دهد. شکل ۸ نمودار های حاصل از اجرای چهار بار تابع MLP است.

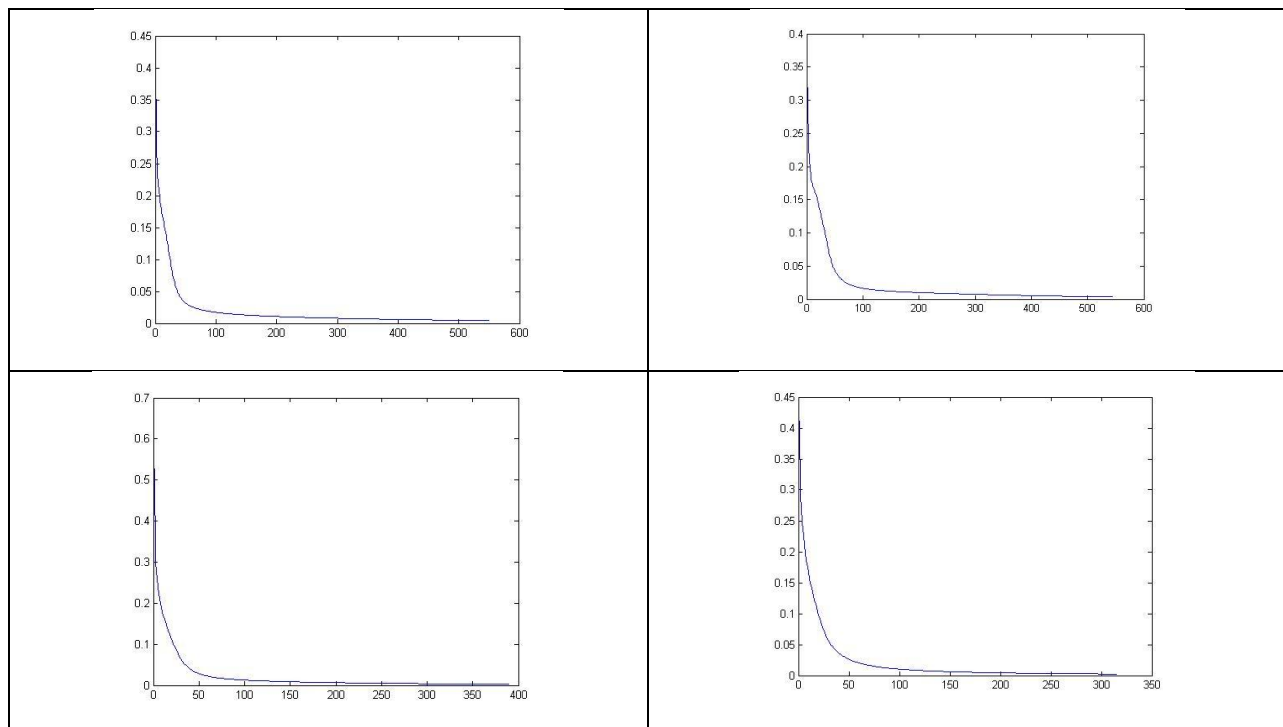
با اجرای تابع MAIN روی مجموعه داده ای iris با نرخ آموزش ۰.۳ نتایج در جدول ۷ حاصل شد. ابتدا پس از جدا کردن داده های تست از آموزش، داده های باقی مانده برای آموزش به چهار قسمت تقسیم می شوند و هر بار سه قسمت برای آموزش و قسمت دیگر برای داده ی اعتبار سنجی به کار گرفته می شوند.

نسل	۶۵۵	۶۵۶	۵۳۶	۴۲۸
کارایی	۹۶.۴۲۸۶	۸۵.۷۱۴۳	۹۶.۴۲۸۶	۹۲.۸۵۷۱

جدول ۷- اجرای الگوریتم MLP روی مجموعه داده ای iris و $LR = 0.3$ با انتخاب n_2 برابر با ۸ به عنوان n_2 مناسب

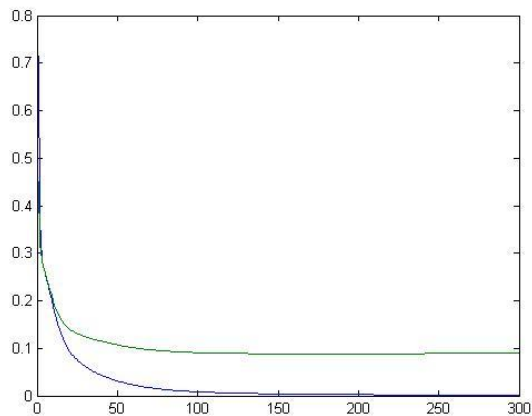


شکل ۷- نمودار حاصل از تابع `earlystoppingmethod` با $LR=0.2$ و $n_2=3$ به عنوان n_2 مناسب

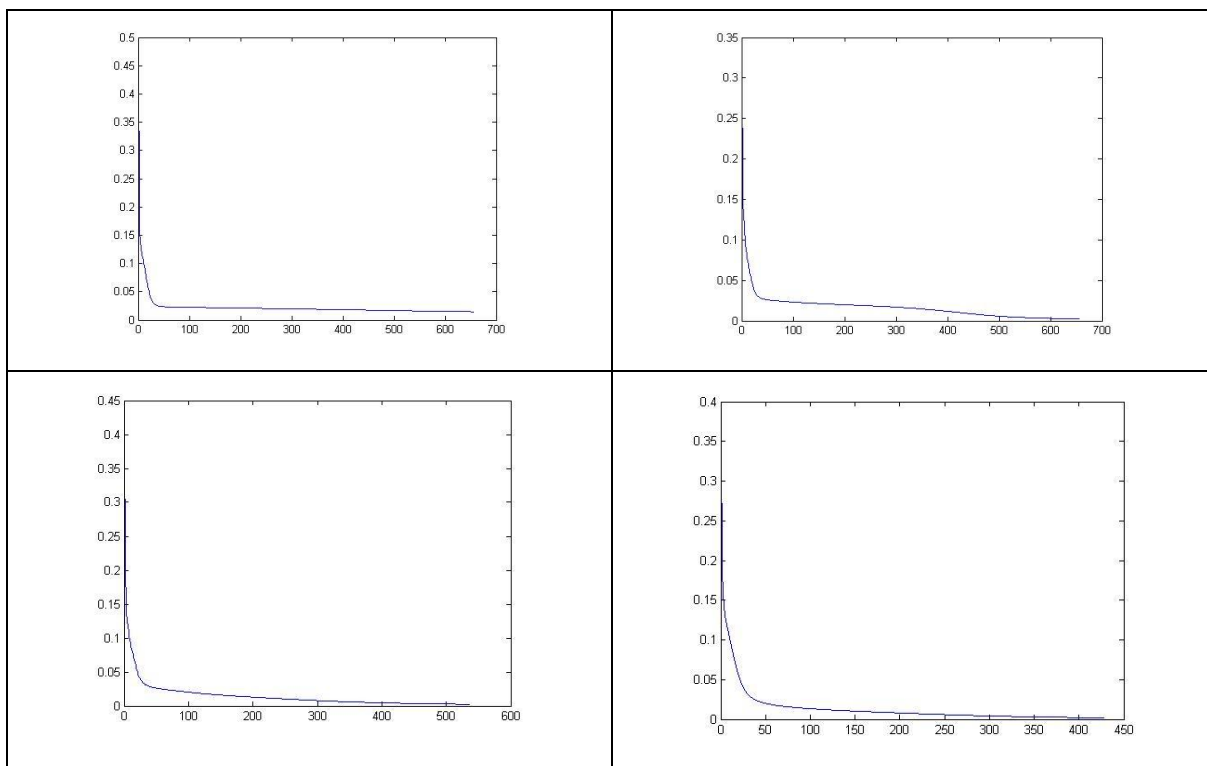


شکل ۸- خطا های حاصل از چهار بار اجرای الگوریتم `MLP` روی مجموعه داده ای `iris` و $LR = 0.2$ با انتخاب n_2 برابر با ۳ به عنوان n_2 مناسب

با توجه به جدول ۷ میانگین و واریانس به ترتیب برابر است با، ۹۲.۸۵۷۱ و ۲۵.۵۱۰۲. شکل ۱۰ نمودار حاصل از تابع `earlystoppingmethod` را نشان می دهد. شکل ۱۱ نمودار های حاصل از اجرای چهار بار تابع `MLP` است.



شکل ۹- نمودار حاصل از تابع `earlystoppingmethod` با $LR=0.3$ و $n_2=8$ به عنوان n_2 مناسب



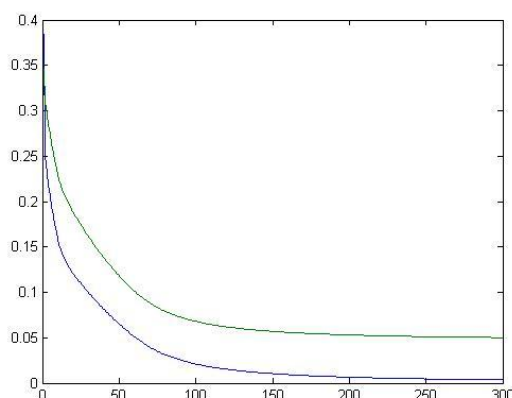
شکل ۱۰- خطاهای حاصل از چهار بار اجرای اجرای الگوریتم `MLP` روی مجموعه داده ای `iris` و $LR = 0.3$ با انتخاب n_2 برابر با ۸ به عنوان n_2 مناسب

با اجرای تابع `MAIN` روی مجموعه داده ای `iris` با نرخ آموزش ۰.۴ نتایج در جدول ۷ حاصل شد. ابتدا پس از جدا کردن داده های تست از آموزش، داده های باقی مانده برای آموزش به چهار قسمت تقسیم می شوند و هر بار سه قسمت برای آموزش و قسمت دیگر برای داده ی اعتبار سنجی به کار گرفته می شوند.

نسل	۱۹۴	۱۵۸	۸۸	۲۱۳
کارایی	۹۶.۴۲۸۶	۸۹.۲۸۵۷	۱۰۰	۹۶.۴۲۸۶

جدول ۷- اجرای الگوریتم MLP روی مجموعه داده ای iris و $LR = 0.4$ با انتخاب n_2 برابر با ۳ به عنوان n_2 مناسب

با توجه به جدول ۷ میانگین و واریانس به ترتیب برابر است با، ۹۵.۵۳۵۷ و ۲۰.۱۹۵۶. شکل ۱۱ نمودار حاصل از تابع **earlystoppingmethod** را نشان می دهد. شکل ۱۲ نمودار های حاصل از اجرای چهار بار تابع MLP است.



شکل ۱۱- نمودار حاصل از تابع **earlystoppingmethod** با $LR=0.4$ و $n_2=3$ به عنوان n_2 مناسب

۵. تحلیل نتایج

به طور کلی بر اساس نتایج به دست آمده، الگوریتم **انتشار بازگشتی** برای طبقه بندی داده ها نسبت به الگوریتم های طبقه بند **اولین نزدیک ترین همسایه**^۱، **طبقه بند بیز**^۲، **K-امین نزدیک ترین همسایه**^۳ و **پنجره ی پارزن**^۴ بهتر عمل می کند.

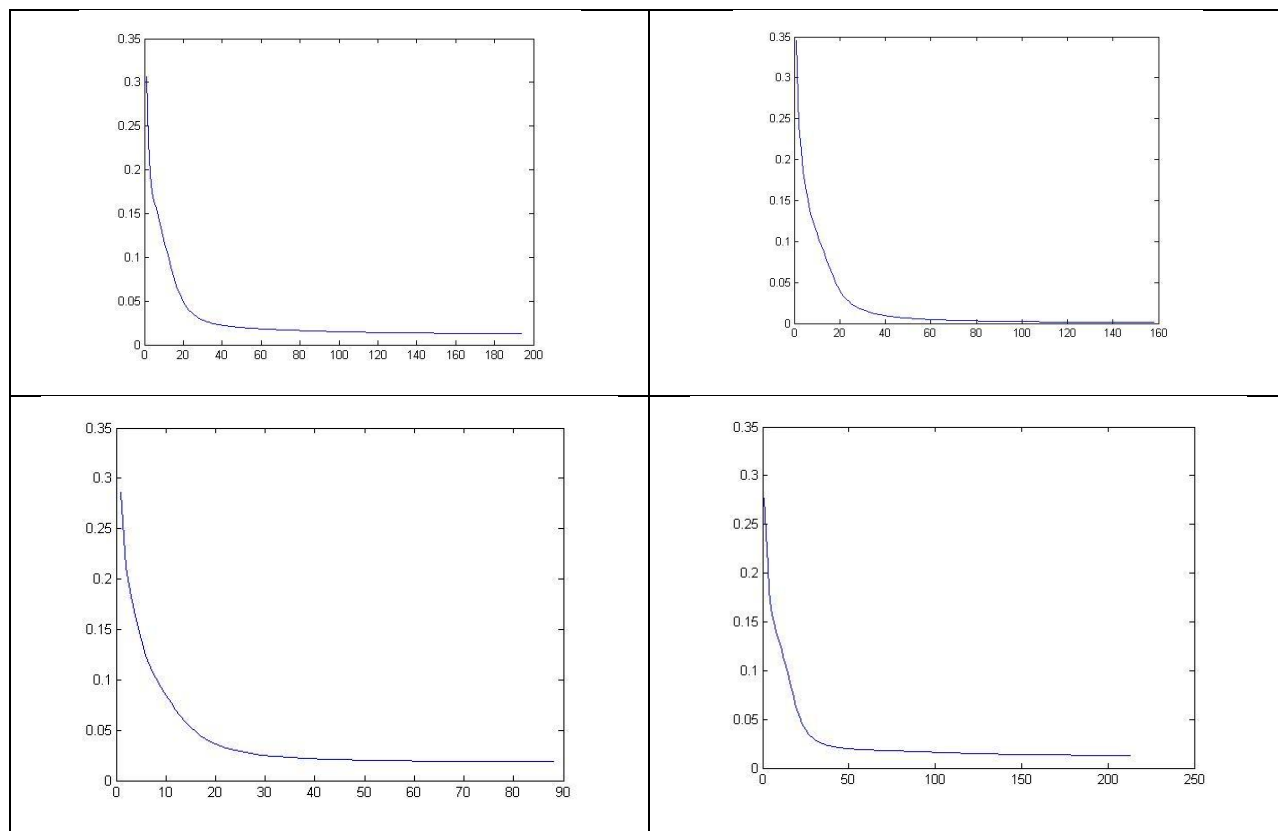
در طی انجام آزمایش ها به این نتیجه رسیده می شود که چنانچه مقدار n_2 و تعداد نسل ها مناسب انتخاب شوند این الگوریتم از کارایی بسیار بالایی برخوردار است.

^۱ One-nearest Neighbor

^۲ Bayes Classifier

^۳ K-nearest Neighbor

^۴ Parzen Window



شکل ۱۲-خطا های حاصل از چهار بار اجرای الگوریتم MLP روی مجموعه داده ای iris و $LR = 0.4$ با انتخاب n_2 برابر با ۳ به عنوان n_2 مناسب