

تکلیف کامپیوتری اول شبکه های عصبی



نیم سال دوم ۱۴۰۲-۱۴۰۳

استاد درس : دکتر محمد فرخی

نویسنده : آرین حاجی زاده

فهرست مطالب

چکیده :	۳
۱- الگوریتم پرسپترون چند لایه	۳
۱-۱- کلاسه بندی برای الگو ماه شکل	۳
۱-۱-۱- معماری شبکه	۴
۲-۱- تخمین تابع غیر خطی صورت سوال	۷
۱-۳-۱- معماری شبکه	۸
۲- پیاده سازی الگوریتم توابع شعاعی پایه (RBF)	۱۲
۱-۲- معماری شبکه	۱۲
۳- نتیجه گیری	۱۶

چکیده:

در این مسئله از ما خواسته شده تا با استفاده از شبکه پرسپترون چند لایه^۱ و در ادامه به صورت اختیاری با استفاده از شبکه توابع شعاعی پایه^۲ رابطه غیر خطی (۱) را مدلسازی کنیم. برای این کار ابتدا الگوریتم پرسپترون چندلایه را برای کلاسه‌بندی اشکال ماه شکل، بررسی خواهیم نمود تا از صحت عملکرد الگوریتم اطمینان حاصل کنیم. سپس با ایجاد تابع خواسته شده به تخمین آن میپردازیم. لازم به ذکر است که تمامی کدها به صورت جدا به پیوست تقدیم خواهد گردید و نیز فایل پروژه با فرمت m-file نیز در انتهای ترم به صورت جداگانه تقدیم می‌گردد.

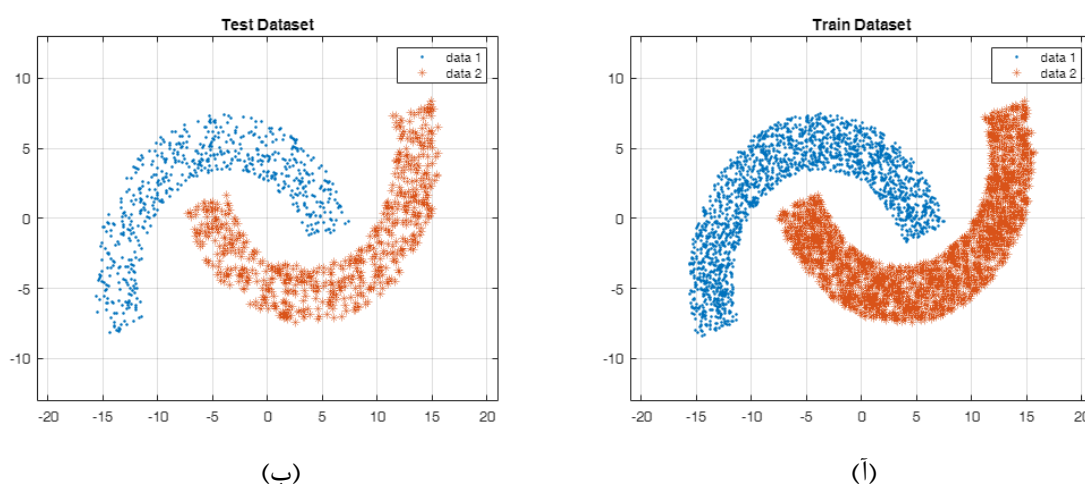
$$y(k) = \frac{y(k-1)y(k-2)y(k-3)+\beta}{1+y^2(k-1)y^2(k-2)} + u(k-1) \quad (1)$$

۱- الگوریتم پرسپترون چند لایه

مطابق مطالب آموزش داده شده در کلاس، ابتدا به پیاده‌سازی اولیه و ساده الگوریتم پرسپترون برای کلاسه‌بندی الگوهای ماه شکل میپردازیم. سپس با اطمینان از عملکرد شبکه در حالت کلی، از معماری بدست آمده برای تخمین تابع داده شده خواهیم پرداخت. این پیاده‌سازی با استفاده از روش بروزرسانی برخط وزن‌ها انجام خواهد گرفت.

۱-۱- کلاسه بندی برای الگو ماه شکل

با مقدمه گفته شده با ایجاد الگوهای ماه شکل با استفاده از قطعه کد موجود در سایت کلاس درس، پیاده سازی را آغاز میکنیم. خروجی این قطعه کد با تنظیمات انجام گرفته در شکل ۱-۱ قابل مشاهده می‌باشد.



شکل ۱-۱- (آ) داده ماه شکل تمرین (ب) داده ماه شکل آزمایش

^۱ Multi-Layer Perceptron

^۲ Radial Basis Functions

حال با استفاده از این الگوها به کلاسه‌بندی می‌پردازیم. نکته بسیار مهم در رابطه با این داده‌ها، این است که بایستی پیش از دادن داده‌ها به الگوریتم، آن‌ها را مطابق با شرایط تابع فعالساز، مقیاس بندی^۳ کنیم. در غیر این صورت، کلاسه بندی به درستی انجام نخواهند شد.

۱-۱-۱- معماری شبکه^۴

با استفاده از یک لایه پنهان و پیروی از قاعده رشد دادن شبکه^۵، با استفاده از ۳ سلول، کلاسه بندی را آغاز میکنیم که در جدول انتهایی فصل تمامی مراحل قابل مشاهده خواهند بود. برای این تکلیف، قاعده متوقف کردن الگوریتم، استفاده از اعتبار سنجی متقابل^۶ داده‌ها، به نحوی که این عمل در هر ۳ دوره یک بار انجام گیرد، میباشد. در این ضابطه تمامی داده‌ها را به دو بخش تمرین و اعتبار سنجی تقسیم میکنیم تا در طول اجرای الگوریتم در صورتی که الگوریتم به همگرایی در وزن‌های تمرین و اعتبار سنجی رسید، در صورتی که خطا به اندازه کافی پایین بود، الگوریتم را متوقف کنیم. در این تکلیف، ضریب تکانه^۷ تنها به جمله پیشین و با مقدار ثابت اعمال میگردد. عملکرد الگوریتم بازای ضرایب تکانه متفاوت، در جدول انتهایی فصل آورده شده است. مقدار ضریب آموزش نیز ثابت بوده و از قاعده دلتا-دلتا یا قاعده دلتا-بار-دلتا پیروی نشده است. لازم به ذکر است، در این گزارش از تابع فعالسازی تانژانت هیپربولیک (رابطه ۱.۱) استفاده شده است. بنابراین پس انتشار خطا به صورت رابطه (۱.۲) و رابطه (۱.۳) که برای مرحله آخر پس انتشار خطا میباشد، به کار برده شده اند.

$$\varphi(v) = \frac{1-e^{-v}}{1+e^{-v}} \rightarrow \varphi'(v) = (1 - \varphi(v)^2) \quad (1.1)$$

$$\Delta W_{ji}(n) = \eta \delta_j y_i + \alpha \Delta W(n-1) \quad (1.2)$$

$$\delta_j(n) = \varphi'_j(v_{j(n)}) \sum_{k=1}^m w_{kj}(n) \delta_k(n) \quad (1.3)$$

در این حالت دسته بندی و نیز میانگین مربعات خطا برای این شبکه در شکل ۱-۲ گزارش شده اند. همانطور که از این شکل پیداست، عملکرد این شبکه در نهایت موفق بوده اما کند به نتیجه رسیده است و تغییرات آن زیاد است. برای تسریع عملکرد شبکه میتوان ضریب آموزش را تغییر داد. اما هنگام افزایش ضریب آموزش، شبکه مذکور پس از بالابردن بیش از مقدار خاصی به اشباع خواهد رسید. بنابراین برای بهبود عملکرد شبکه تعداد سلول‌های لایه پنهان را به ۱۰ افزایش میدهیم و برای کاهش نوسانات، بایستی

^۳ Scale

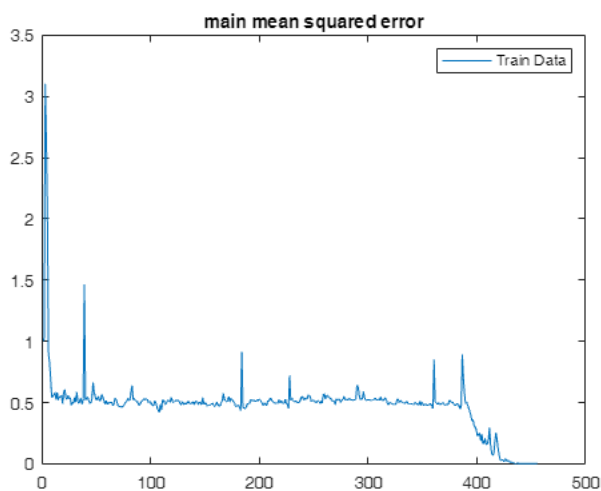
^۴ Network Architecture

^۵ Network Growing

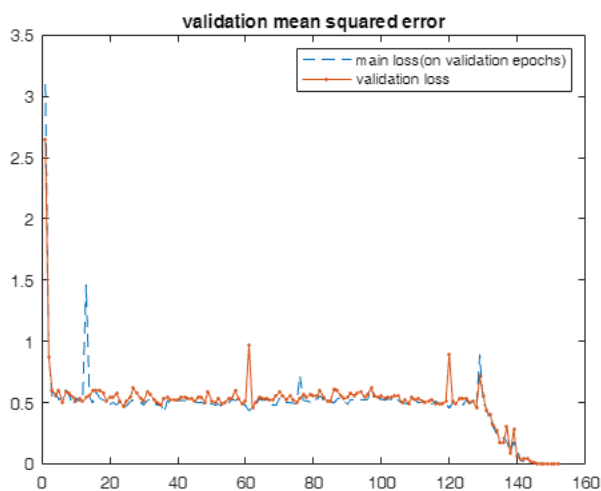
^۶ Cross Validation

^۷ Momentum

مقدار ضریب آموزش یا ضریب تکانه را کاهش دهیم. با انجام نکات بالا بار دیگر خروجی را در شکل ۳-۱ مشاهده میکنیم.

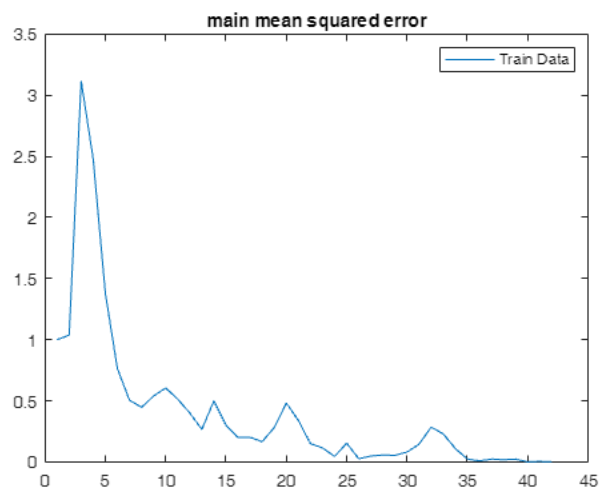


شکل ۲-۱- (آ) میانگین مربعات خطا برای شبکه مذکور

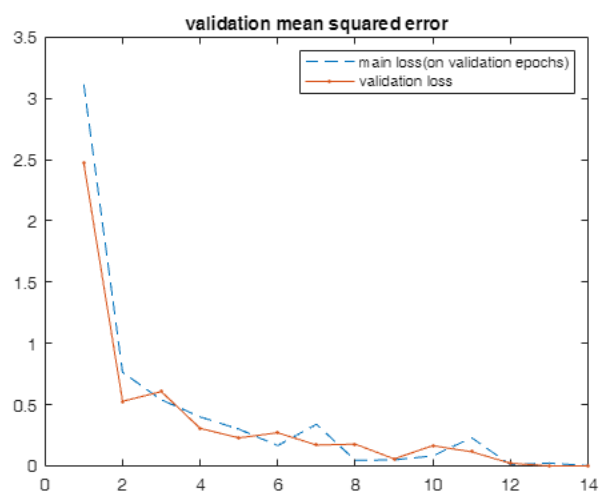


شکل ۲-۱- (ب) مقایسه میانگین مربعات خطا در داده اصلی و اعتبار سنجی در دوره‌های منتخب

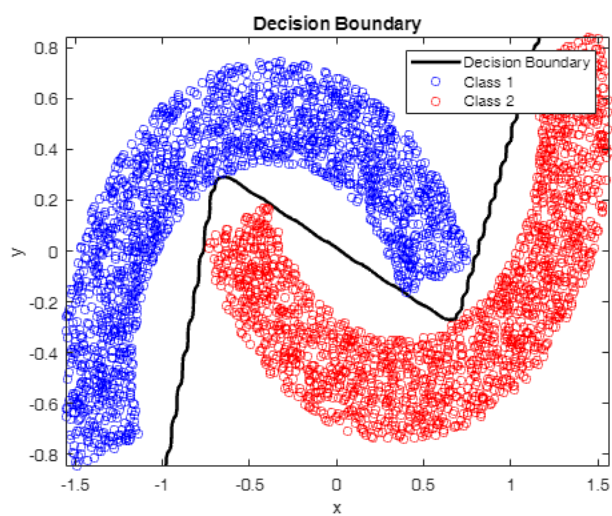
با انجام پیشنهادات ذکر شده، حال شبکه نهایی را بدست خواهد آمد. نتایج نهایی بدست آمده از این شبکه، در شکل ۳-۱ قابل مشاهده می‌باشند.



شکل ۱-۳- (آ) میانگین مربعات خطا برای شبکه مذکور



شکل ۱-۳- (ب) مقایسه میانگین مربعات خطا در داده اصلی و اعتبار سنجی در دوره‌های منتخب



شکل ۱-۳- (پ) جداسازی انجام گرفته

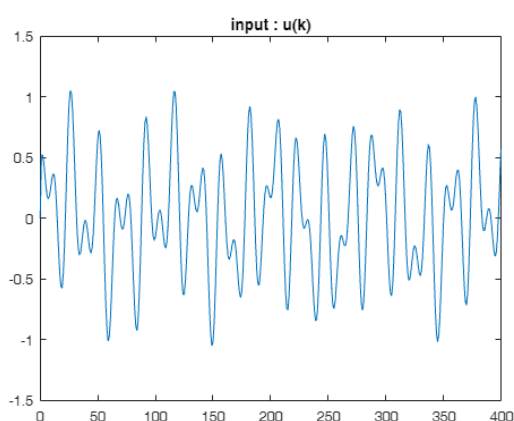
همانطور که از این شکل پیداست الگوریتم پرسپترون چند لایه با موفقیت در این مثال قادر به جداسازی دو الگو بوده. لازم به ذکر است به علت مقدار دهی تصادفی اولیه وزن‌ها جداسازی نیز به صورت تصادفی انجام خواهد گرفت و به همین علت در دو اجرای متوالی، احتمال امکان مشاهده دو شکل یکسان، بسیار پایین می‌باشد. نتایج شبکه بازای تعداد سلول‌های بیشتر، در جداول انتهایی آورده خواهد شد. با انجام این مرحله و حاصل شدن اطمینان از صحت عملکرد شبکه، به خواسته اصلی سوال می‌پردازیم.

۱-۲- تخمین تابع غیر خطی صورت سوال

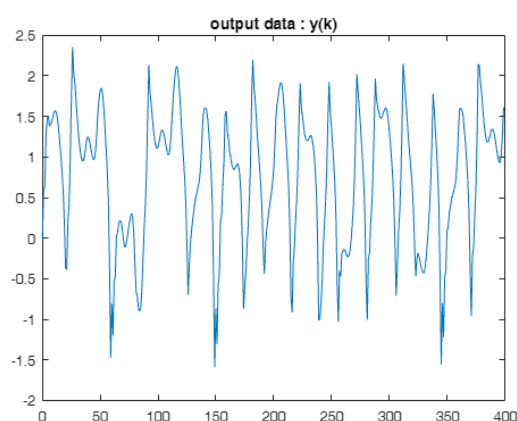
برای انجام این بخش ابتدا بایستی ورودی و خروجی تابع را با استفاده از مقادیر و ضوابط داده شده در صورت سوال، پیاده سازی نماییم. با توجه به اختیاری بودن انتخاب، مقدار β برابر با ۱.۱ در نظر گرفته شده است. بنابراین ورودی تابع را با استفاده از ضابطه داده شده به شکل زیر می‌نویسیم.

$$u(k) = 0.5 \sin\left(\frac{\pi k}{11}\right) + 0.4 \cos\left(\frac{\pi k}{6.5}\right) + 0.2 \sin\left(\frac{\pi k}{45}\right) \quad (۱.۴)$$

خروجی این تابع نیز در رابطه ۱ نشان داده شده بود. ورودی و خروجی این تابع را در شکل ۱-۴ مشاهده می‌نماییم.



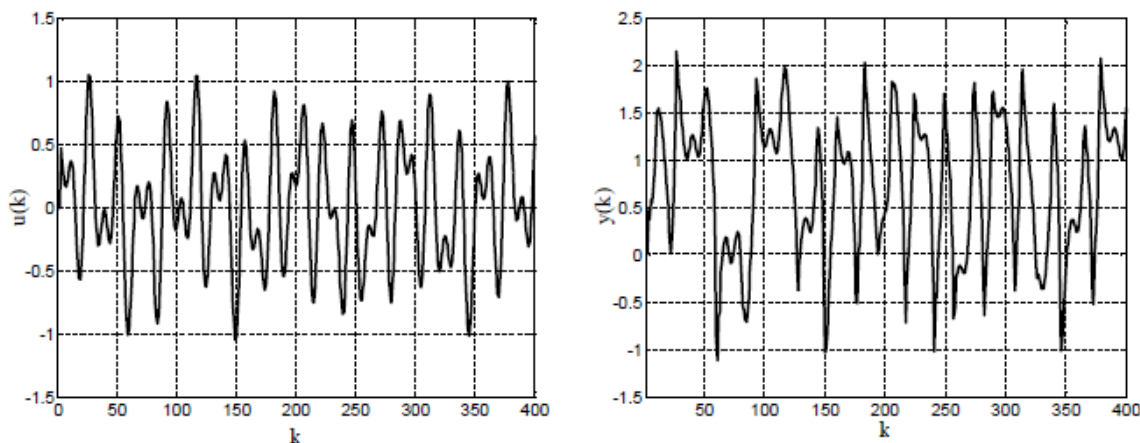
(ب) ورودی داده شده



(آ) خروجی داده شده

شکل ۱-۴- ورودی و خروجی داده شده

نتایج بدست آمده را با اشکال داده شده در صورت سوال مقایسه می‌کنیم.

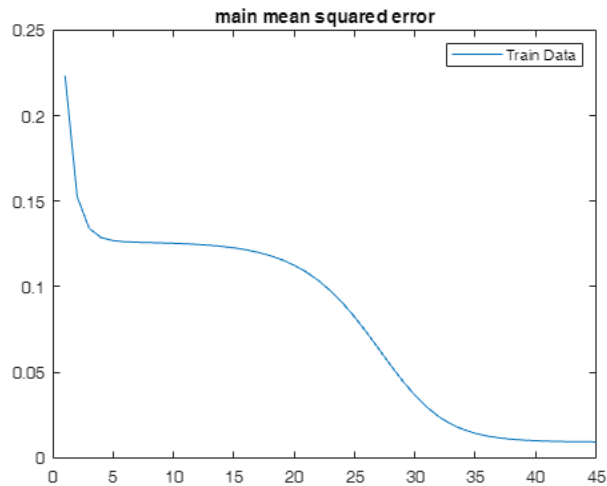


شکل ۱-۵- ورودی و تابع داده شده در صورت سوال

۱-۳-۱- معماری شبکه

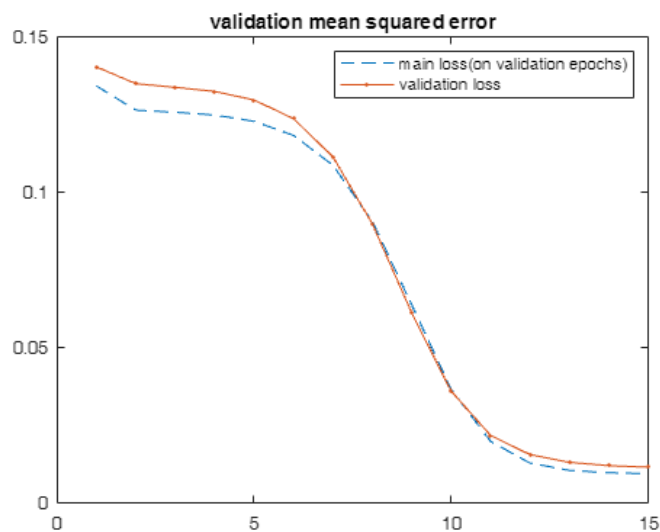
در این بخش نیز مانند معماری شبکه برای کلاسه بندی الگوهای ماه شکل عمل شده. اما تفاوت عمده این بخش با بخش کلاسه بندی الگوهای ماه شکل نوع مسئله می‌باشد. جنس این مسئله از نوع تخمین تابع و جنس مرحله پیشین، کلاسه بندی می‌باشد. برای تخمین تابع با توجه به قضیه کولمگروف، میتوان با حداکثر دو لایه پنهان تخمین را انجام داد. اما با توجه به قضیه تخمین عمومی یک لایه پنهان با تعداد سلول کافی برای تخمین تابع کافی می‌باشد. برای این مسئله نیز از رشد شبکه استفاده کرده و با یک لایه پنهان و تعداد محدودی سلول شروع میکنیم و شبکه را گسترش می‌دهیم. مقدار نهایی سلول‌ها برای این مسئله برابر ۳ می‌باشد. ضمناً لازم به ذکر است که ضریب آموزش نهایی، برابر ۰.۰۰۵ و ضریب تکران نیز برابر ۰.۰۰۰۵ انتخاب شده اند (نتایج ضرایب دیگر در جداول پایان بخش آمده است). برای آموزش شبکه داده‌ها تماماً مقیاس بندی شده‌اند تا با استفاده از تابع فعالسازی تانژانت هیپربولیک بتوان آموزش لازم را انجام داد. علاوه بر این، داده‌های ساخته شده، به طور تصادفی به داده‌های تمرین و داده‌های اعتبار سنجی با نسبت ۸۰ بر ۲۰ تقسیم شده‌اند.

با این مقدمه حال به بررسی عملکرد نهایی شبکه داده شده برای تخمین تابع مذکور می‌پردازیم. میانگین مربعات خطا برای این الگوریتم و تابع مورد نظر در شکل ۱-۶ نشان داده شده است.



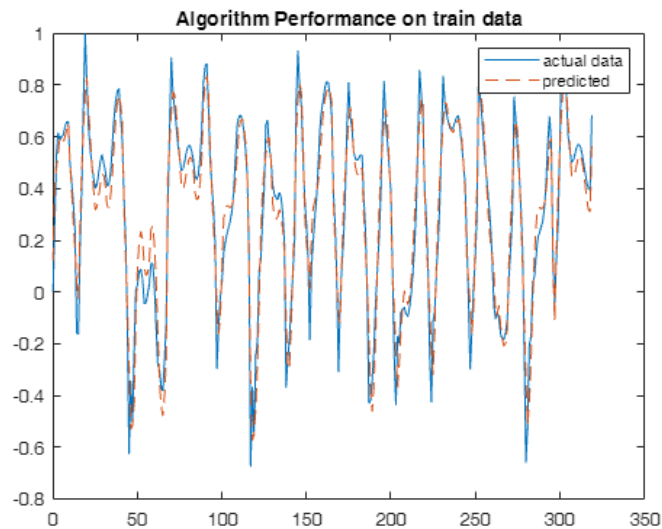
شکل ۶-۱- میانگین مربعات خطا

در این بخش نیز مانند بخش پیشین از اعتبار سنجی متقابل برای متوقف کردن الگوریتم استفاده شده است. با اینکه تعداد دوره‌ها برابر ۱۰۰ تعیین شده بود، همانطور که از شکل نیز پیداست، با این ضابطه توقف، الگوریتم در ۴۵ دوره توقف پیدا کرده. میانگین مربعات خطای اعتبار سنجی متقابل نیز، در شکل ۷-۱ قابل مشاهده می‌باشد.

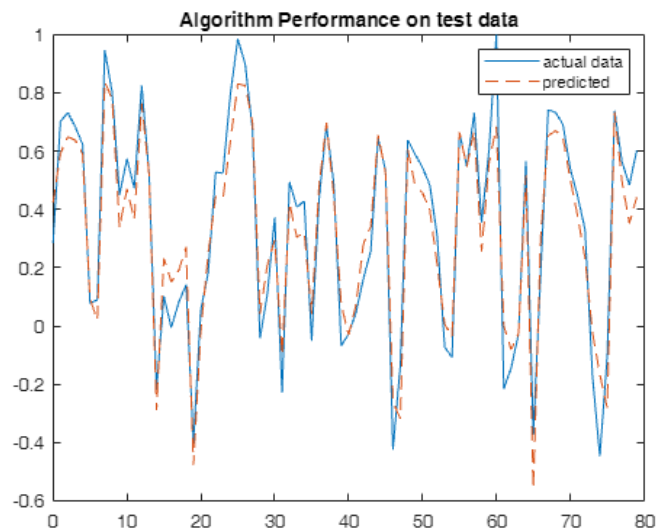


شکل ۷-۱- میانگین مربعات خطا داده‌های تمرین و اعتبار سنجی در دوره‌های منتخب

با نتایج بدست آمده، انتظار می‌رود شبکه عملکرد موفق‌تری در تخمین تابع داشته باشد. حال برای صحت سنجی این سخن، خروجی تابع را مشاهده می‌کنیم.



شکل ۱-۸- تخمین صورت گرفته روی داده‌های تمرین



شکل ۱-۹- تخمین صورت گرفته روی داده‌های اعتبار سنجی

همانطور که مشاهده شد، شبکه طراحی شده به خوبی و با دقت بالایی قادر به تخمین تابع داده شده، با وجود تعداد کم داده‌های ورودی بود. این یکی از ویژگی‌های الگوریتم پرسپترون چند لایه است که قادر به عملکرد مناسب با تعداد داده‌های کم می‌باشد. اقداماتی می‌تواند صورت گیرد که در پیاده سازی‌های بعدی عملکرد شبکه را از حال موجود نیز بهینه تر سازد. از جمله این بهبود ها میتوان به اعمال ضریب تکانه بر روی تمام داده‌های پیشین، پیاده سازی الگوریتم دلتا-دلتا یا دلتا-بار-دلتا و یا بکار بردن روش نیوتن یا استفاده از روش لونبرگ-مارکارت اشاره نمود. اما برای این گزارش به دلیل محدودیت زمانی موجود، سعی شد الگوریتم پرسپترون چندلایه، به صورت ساده پیاده سازی شود.

پس از تخمین این تابع با استفاده از الگوریتم پرسپترون چندلایه، در قسمت بعدی به تخمین تابع با استفاده از الگوریتم توابع شعاعی پایه خواهیم پرداخت. اما ابتدا نتایج شبیه سازی بازای پارامترهای متفاوت را در جداول زیر مشاهده می‌کنیم.

جدول ۱-۱- مشاهده خروجی شبکه برای ضرایب تکانه مختلف ، ضریب آموزش ۰.۰۰۰۵ و تعداد سلول ۳۰

وضعیت	میانگین مربعات خطا	تعداد دوره	ضریب تکانه	شماره
پایدار	۰.۰۱	۱۰۰	۰	۱
پایدار	۰.۰۰۹۸	۱۰۰	۰.۰۰۰۱	۲
پایدار	۰.۰۰۹۴	۴۵	۰.۰۰۰۵	۳
پایدار	۰.۰۰۸۷	۱۰۰	۰.۰۰۵	۴
ناپایدار	NAN	۱۰۰	۰.۰۵	۵

جدول ۱-۲- مشاهده خروجی شبکه برای ضرایب آموزش مختلف ، ضریب تکانه ۰.۰۰۰۰۵ و تعداد سلول ۳۰

وضعیت	میانگین مربعات خطا	تعداد دوره	ضریب آموزش	شماره
پایدار	۰.۱۵	۱۰۰	۰.۰۰۰۰۵	۱
پایدار	۰.۱۳	۱۰۰	۰.۰۰۰۱	۲
پایدار	۰.۱۲	۱۰۰	۰.۰۰۰۵	۳
پایدار	۰.۱۰	۱۰۰	۰.۰۰۱	۴
پایدار	۰.۰۰۹	۱۰۰	۰.۰۰۵	۵
پایدار	۰.۰۰۸۶	۱۰۰	۰.۰۱	۶
اشباع	۰.۶۱	۱۰۰	۰.۰۵	۷
اشباع	۰.۶۱	۱۰۰	۰.۱	۸

جدول ۱-۳- مشاهده خروجی شبکه برای تعداد سلول متفاوت و ضرایب آموزش ۰.۰۰۰۵ و تکانه ۰.۰۰۰۰۵

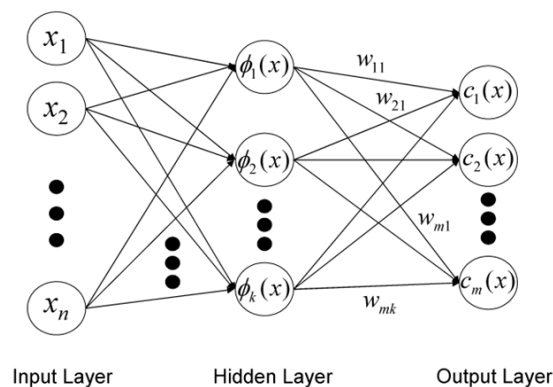
وضعیت	میانگین مربعات خطا	تعداد دوره	تعداد سلول	شماره
پایدار	۰.۰۰۸	۱۰۰	۱	۱
پایدار	۰.۰۰۷	۱۰۰	۲	۲
پایدار	۰.۰۰۷	۱۰۰	۳	۳
پایدار	۰.۰۰۸	۱۰۰	۵	۴
پایدار	۰.۰۰۷	۱۰۰	۱۰	۵
پایدار	۰.۰۰۹	۱۰۰	۲۰	۶
پایدار	۰.۰۰۹	۱۰۰	۳۰	۷
پایدار	۰.۰۰۹	۱۰۰	۴۰	۸

همانطور که از دو جدول بالا مشخص است باید به مقادیر ضرایب آموزش و تکانه توجه ویژه‌ای شود. با افزایش ضریب آموزش اگر مقدار ضریب تکانه مناسب انتخاب شود از ناپایداری شبکه جلوگیری خواهد شد ولی شبکه به اشباع می‌رود اما با انتخاب ضریب تکانه اشتباه هرچند ضریب آموزش مناسبی داشته باشیم، شبکه می‌تواند ناپایدار شود. همانطور که مشاهده می‌شود پس از مقدار مشخصی خطای نهایی تفاوت چندانی نخواهد کرد و برای این مسئله تنها تعداد محدودی سلول جوابگو خواهد بود و نیازی به ایجاد یک شبکه پیچیده نیست. اما برای مسئله کلاسه بندی الگوهای ماه شکل پیچیدگی شبکه بیشتر است و حساسیت زیادی به تغییر تک تک متغیرها دارد.

توجه: به علت مقدار دهی تصادفی اولیه وزن‌ها ممکن است در اجراهای پیاپی اعداد گزارش شده در جدول نتایج متفاوتی داشته باشند. همچنین اعداد گزارش شده به صورت گرد شده می‌باشند و اعداد یکسان در بخش‌های مختلف ممکن است تفاوت اندکی با یکدیگر داشته باشند. نکته آخر مقیاس بندی خطا می‌باشد زیرا با مقیاس بندی داده‌های ورودی خطا نیز مقیاس بندی می‌شود و خطا باید در مجذور اندازه بیشترین مقدار داده اولیه ضرب شود.

۲- پیاده سازی الگوریتم توابع شعاعی پایه (RBF)

پس از پیاده سازی الگوریتم پرسپترون چند لایه برای تخمین تابع مورد نظر حال با استفاده از الگوریتم توابع شعاعی پایه به تخمین خواهیم پرداخت.



شکل ۲-۱- ساختار کلی شبکه RBF

۲-۱- معماری شبکه

در این شبکه معماری مورد نظر بکارگیری الگوریتم ترکیبی بدون نظارت (k میانگین)^۸ برای لایه پنهان و الگوریتم با نظارت کمترین مربعات بازگشتی^۹ برای پیدا کردن وزن‌ها از لایه پنهان به لایه خروجی، استفاده گردیده است. تابع استفاده شده به عنوان تابع گرین پایه، همان تابع گاوسی (رابطه ۲.۱) می‌باشد. همچنین برای الگوریتم کمترین مربعات بازگشتی از روابط (۲.۲) تا (۲.۶) استفاده شده است.

^۸ K-means

^۹ Recursive Least Squares

$$G = e^{-\frac{\|x-t_l\|}{2\sigma^2}} \quad (2.1)$$

$$P(n) = R^{-1}(n) \quad (2.2)$$

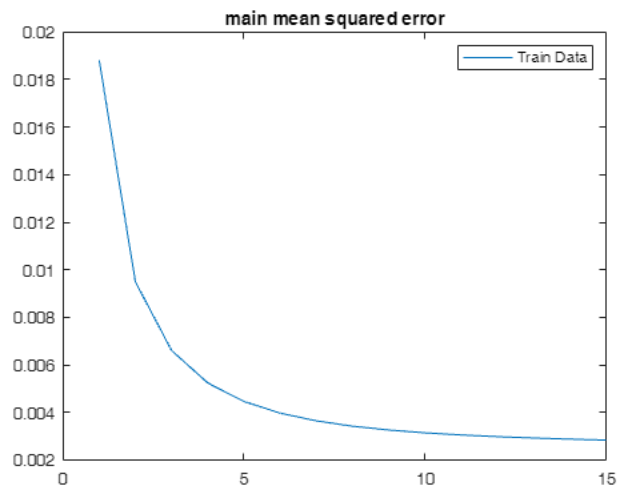
$$P(n) = P(n-1) - \frac{(P(n-1)\varphi(n)\varphi^T(n)P(n-1))}{1+(\varphi^T(n)P(n-1)\varphi(n))} \quad (2.3)$$

$$g(n) = \Phi(n)P(n) \quad (2.4)$$

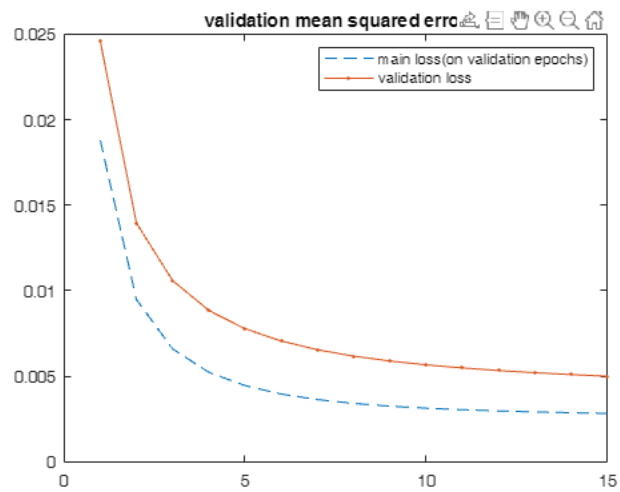
$$\alpha(n) = d(n) - \hat{w}^T(n-1)\Phi(n) \quad (2.5)$$

$$\hat{w}^T(n) = \hat{w}^T(n-1) + g(n)\alpha(n) \quad (2.6)$$

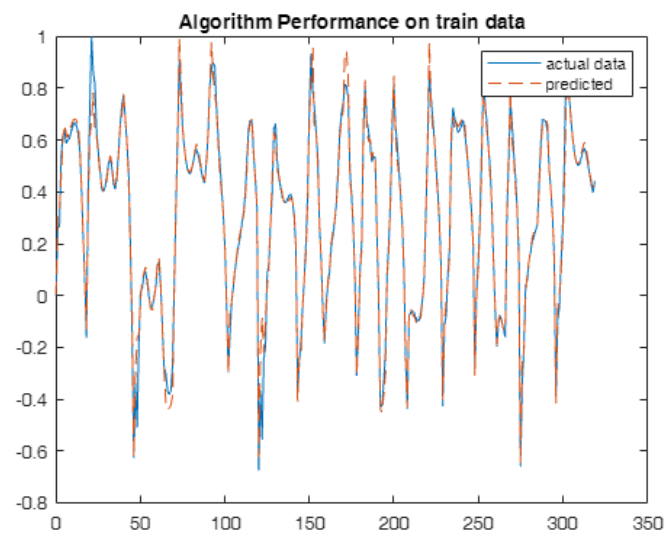
همچنین باید توجه شود که مقدار اولیه P با استفاده از $\lambda I, \lambda = 0,8$ تعیین شده. با استفاده از روابط، بالا حال به پیاده سازی الگوریتم می‌پردازیم. لازم به ذکر است، برای استفاده از روش k میانگین از منابع گوناگونی کمک گرفته شده که در بخش منابع قابل مشاهده خواهند بود. با استفاده از روابط بالا و به کارگیری سلول پنهان ($k = 40$) و یک بایاس توانستیم به نتایج بسیار خوبی دست پیدا کنیم. که در شکل‌های ۲-۲ تا ۲-۵ این نتایج قابل مشاهده می‌باشند. در این پیاده سازی نیز برای توقف الگوریتم از اعتبار سنجی متقابل استفاده شده با این تفاوت که اختلاف میانگین مربعات خطای داده تمرین و اعتبار سنجی کمی بیشتر در نظر گرفته شده است. زیرا داده تمرین، خطای بسیار پایینی دارد و ممکن است داده اعتبار سنجی دقیقاً به مقدار آن نرسد.



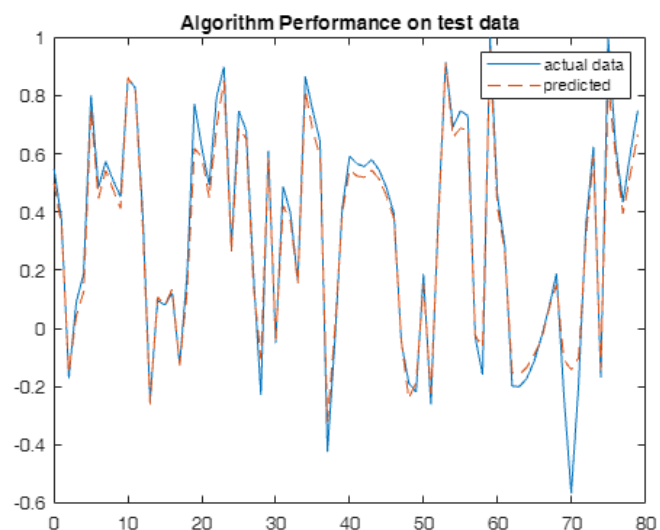
شکل ۲-۲- میانگین مربعات خطا



شکل ۲-۳- میانگین مربعات خطا داده های تمرین و اعتبار سنجی در دوره های منتخب



شکل ۲-۴- عملکرد شبکه RBF بر روی داده های تمرین



شکل ۲-۵- عملکرد شبکه RBF بر روی داده‌های اعتبار سنجی

جدول ۲-۱- عملکرد شبکه بازای K های مختلف

وضعیت	میانگین مربعات خطا	تعداد دوره	مقدار K	شماره
پایدار	۰.۱۲	۱۰۰	۱	۱
پایدار	۰.۰۶	۱۰۰	۲	۲
پایدار	۰.۰۵	۱۰۰	۳	۳
پایدار	۰.۰۲	۱۰۰	۵	۴
پایدار	۰.۰۰۸	۵۳	۱۰	۵
پایدار	۰.۰۰۶	۵۰	۲۰	۶
پایدار	۰.۰۰۴	۱۰	۳۰	۷
پایدار	۰.۰۰۱	۱۰	۴۰	۸

جدول ۲-۲- عملکرد شبکه بازای λ های مختلف $k=20$

وضعیت	میانگین مربعات خطا	تعداد دوره	مقدار λ	شماره
پایدار	۰.۰۰۶	۶۳	۰.۰۱	۱
پایدار	۰.۰۰۷	۶۵	۰.۰۵	۲
پایدار	۰.۰۰۴	۸۴	۰.۱	۳
پایدار	۰.۰۰۵	۴۴	۰.۵	۴
پایدار	۰.۰۰۴	۷	۱	۵
پایدار	۰.۰۰۴	۶	۳	۶
پایدار	۰.۰۰۵	۱۰	۵	۷
پایدار	۰.۰۰۶	۸	۱۰	۸
پایدار	۰.۰۰۵	۷	۱۰۰	۹

با توجه به این مشاهدات در درجه اول تعداد K یا همان درجه آزادی شبکه برای دقت آن بسیار مهم است به طوری که با بالاتر رفتن درجه آزادی دقت شبکه به طور چشم گیری افزایش پیدا می کند. اما در مورد مقدار λ میتوان گفت با افزایش مقدار آن، شبکه افزایش سرعت پیدا می کند. اما این افزایش پس از نقطه مشخصی دیگر تغییری در روند و همگرایی شبکه نمی دهد.

۳- نتیجه گیری

همانطور که از شکل های هر دو بخش پیشین پیداست، هر دو شبکه عملکرد مناسبی از خود نشان دادند. اما میتوان گفت شبکه RBF بسیار سریع تر از MLP به نتیجه رسید همچنین نتیجه بدست آمده، دارای دقت بیشتری می باشد. برای شبکه MLP باید برای آغاز الگوریتم بسیار دقت داشته باشیم تا ورودی ها به شکل درست به الگوریتم داده شوند. در غیر این صورت با دادن داده های مقیاس بندی نشده و نیز داده های با تنوع پایین شبکه عملکرد مناسبی نخواهد داشت. به عنوان مثال دو اشکال ذکر شده در حین پیاده سازی باعث اشباع شدن تابع و یا در حتی در صورت پایان عملیات خراب شدن تخمین خواهند شد. در شبکه RBF برای بدست آوردن دقت های بالا باید توجه داشت که تعداد مراکز مناسب انتخاب شوند در صورت تعداد پایین مراکز الگوریتم قادر به رسیدن به دقت مناسب نخواهد بود. دیگر نکته حائز اهمیت درباره این الگوریتم انتخاب صحیح مقدار λ میباشد. لازم به توجه است که برای الگوریتم RBF امکان ماندن در مینیمم محلی وجود دارد و ممکن است نیاز به اجرای چند باره الگوریتم برای رسیدن به دقت کافی باشد.

به طور کلی هر دو الگوریتم در این مسئله پاسخ مناسبی داشتند. الگوریتم RBF عملکرد سریع تری داشت و در تکرارهای کمتری به دقت خواسته شده رسید اما الگوریتم MLP با تعداد تکرار بیشتری توانست به دقت مشابه برسد. الگوریتم MLP با تعداد سلول های پایین، در تکرارهای بالا قادر به رسیدن به دقت لازم بود اما برای الگوریتم RBF این مسئله برقرار نیست و با تعداد سلول های پایین همچنان بسیار سریع همگرا می شود اما این همگرایی دقت لازم را ندارد و برای دقت های بالاتر باید درجه آزادی شبکه یعنی مقدار K را افزایش دهیم. همچنین در صورت نداشتن داده کافی الگوریتم MLP عملکرد بهتری نسبت به RBF خواهد داشت.