

مقایسه توابع یادگیری شبکه عصبی در مدل سازی رواناب

محمدجواد زینلی^۱، سید رضا هاشمی^{۲*}

۱. دانشجوی دکتری منابع آب، گروه علوم و مهندسی آب، دانشگاه بیرجند

۲. استادیار گروه علوم و مهندسی آب، دانشگاه بیرجند

(تاریخ دریافت ۱۳۹۵/۰۹/۰۱؛ تاریخ تصویب ۱۳۹۵/۱۰/۱۰)

چکیده

پیش‌بینی دقیق جریان در رودخانه‌ها یکی از ارکان مهم در مدیریت منابع آب‌های سطحی به‌ویژه اتخاذ تدابیر مناسب در مواقع سیلاب و بروز خشکسالی‌هاست. درحقیقت، حصول روش‌های مناسب و دقیق در پیش‌بینی جریان رودخانه‌ها را می‌توان به‌عنوان یکی از چالش‌های مهم در فرایند مدیریت و مهندسی منابع آب دانست؛ اگرچه تحقیقات وسیعی در خصوص کاربرد روش‌های متکی بر شبکه‌های عصبی مصنوعی دقت این روش‌ها بر روش‌های متداول آماری مانند روش‌های اتورگسیو و میانگین متحرک ارائه شده است. در این تحقیقات برای یافتن بهترین ساختار برای شبکه عصبی تنها به تغییر تعداد لایه‌های پنهان و تعداد نورون‌ها اکتفا می‌شود و به‌دلیل پیچیدگی حاکم بر انتخاب و معماری شبکه مناسب، استفاده از آنها در عمل به‌طور مناسب توسعه نیافته است. در این تحقیق تعداد ۱۵ تابع یادگیری در شبکه عصبی بررسی شد و نتایج نشان داد در ساختار شبکه با یک لایه پنهان (ANN1) تابع یادگیری $\text{learnl}v1$ ، learnh و learnis به‌ترتیب با MSE برابر $0/000158$ ، $0/000185$ و $0/000188$ و در مدل ساختار شبکه با دو لایه پنهان ANN2 توابع یادگیری learnh ، learnsomb و learncon به‌ترتیب با MSE برابر $0/000154$ ، $0/000173$ و $0/000176$ ، عملکرد مناسب‌تری نسبت به دیگر توابع یادگیری داشته‌اند. از سوی دیگر در ده مرتبه اجرای دو مدل، دو تابع یادگیری learnsom و learnngdm در مدل ANN1 و learnh و learnos در مدل ANN2، بیشترین تکرار را در بین بهترین توابع یادگیری، داشته‌اند و بنابراین، هنگام استفاده از شبکه پسانتشار خطا (که تابع یادگیری آن learnngdm است) بهتر است تعداد لایه پنهان بیشتر از یکی نباشد؛ زیرا در این صورت شانس رسیدن به جواب مناسب بیشتر خواهد بود، اما اگر به‌دنبال زیادکردن عملکرد شبکه با افزایش تعداد لایه پنهان باشیم بهتر است با احتیاط از پیش‌فرض شبکه و به‌طور مشخص از learnngdm استفاده شود.

کلیدواژگان: پیش‌بینی، توابع یادگیری، شبکه عصبی مصنوعی، معیار عملکرد.

مقدمه

پدیده بارش- رواناب از پیچیده‌ترین فرایندهای هیدرولوژیکی است که به دلیل پارامترهای زیاد و ناپایداری مشخصه‌های حوضه آبریز شبیه‌سازی آن مشکل است. در دهه‌های اخیر استفاده از شبکه‌های عصبی مصنوعی^۱ در شاخه‌های مختلف مهندسی از جمله مهندسی آب به دلیل قابلیت این روش‌ها در تخمین و شبیه‌سازی توابع غیرخطی مانند بارندگی، رواناب و جریان رودخانه با دقت مناسب رواج یافته است [۱]. شبکه‌های عصبی مصنوعی برخلاف روش‌های تجربی، این قابلیت را دارند که از روی یک سری اطلاعات آموزش ببینند و رابطه مناسبی بین آنها برقرار کنند. بنابراین، در مسائلی که ایجاد ارتباط بین داده‌ها مشکل است، شبکه‌های عصبی مصنوعی بسیار کارآمد هستند [۲]. محققان بسیاری کارایی شبکه عصبی را بررسی کرده‌اند. فتحی و همکارانش (۲۰۰۹) آورد ورودی به سد وحدت سهندج را با استفاده از شبکه عصبی مصنوعی مدل‌سازی کردند. نتایج تحقیق ایشان نشان‌دهنده انطباق خوب بین مقادیر پیش‌بینی‌شده با شبکه عصبی ترکیبی و داده‌های مشاهداتی وجود دارد و دقت مدل دینامیکی بیشتر از مدل استاتیکی است [۳]. دارام (۲۰۱۰) داده‌های بارش رواناب حوضه سوسورلوک^۲ را با شبکه عصبی مصنوعی و سیستم فازی مدل‌سازی کردند. نتایج پژوهش آنها نشان داد مدل فازی و شبکه عصبی نتایج تقریباً مشابهی دارند [۴]. چوا و وانگ (۲۰۱۱) با استفاده از شبکه عصبی مصنوعی و مدل آنالیز اسپکترون منفرد به مدل‌سازی بارش- رواناب پرداختند. نتایج این تحقیق نشان داد شبکه عصبی در مقایسه با آنالیز اسپکترون منفرد عملکرد بهتری دارد [۵]. پاتیل و همکارانش (۲۰۱۲) الگوریتم‌های مختلف پیش‌بینی بارش رواناب برای مصرف بهتر آب برای منطقه نیمه‌خشک از غرب استان مهاراشترا^۳ که پایه کشاورزی هند است را بررسی کردند [۶].

در این تحقیق ۱۵ مورد از توابع یادگیری شبکه عصبی، پیش‌بینی رواناب حوضه سد ملاصدرا بررسی شده است. هدف از این تحقیق مقایسه بین توابع یادگیری شبکه عصبی مصنوعی در پیش‌بینی رواناب است. بدین منظور از داده‌های روزانه بارش، دما، تبخیر و رواناب

استفاده شده است. ساختار شبکه عصبی با یک لایه پنهان (ANN1) و دو لایه پنهان (ANN2) در پنج اجرا بررسی و در نهایت بهترین توابع یادگیری شبکه عصبی مصنوعی مشخص می‌شود.

مواد و روش‌ها

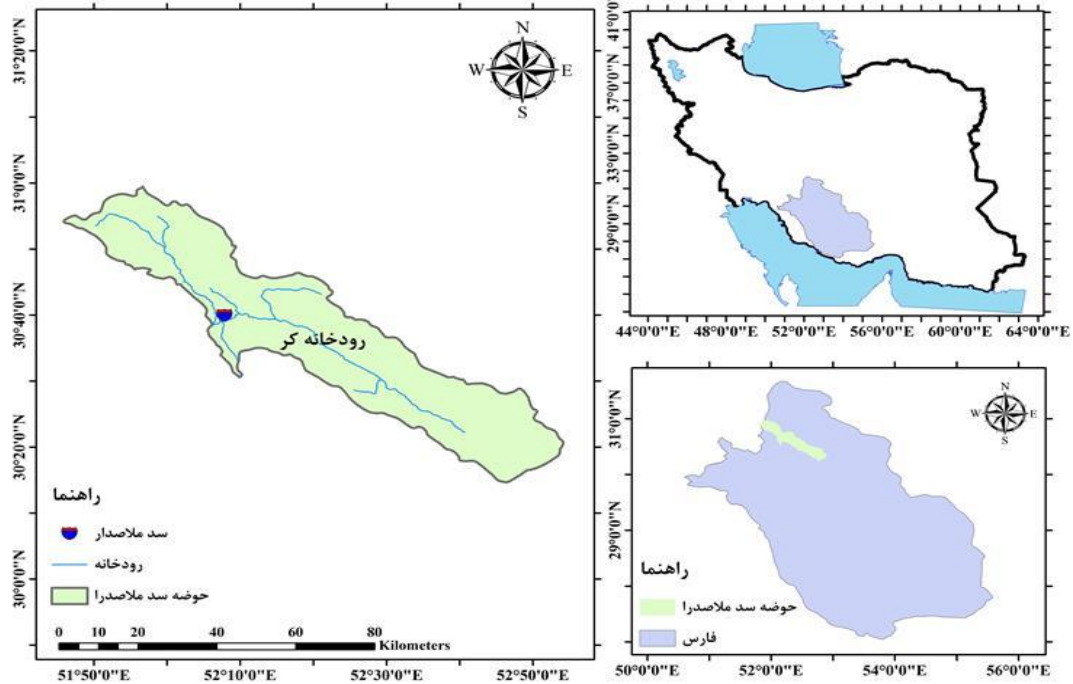
منطقه مطالعه شده

حوضه سد ملاصدرا جزئی از حوضه فرعی طشک، بختگان و مهارلو و مساحت آن حدود ۲۲۵۰ کیلومترمربع است. رودخانه کر از رودهای مهم و پرآب استان فارس به حساب می‌آید؛ این رودخانه از به‌هم‌پیوستن رودخانه تنگ براق و شور شیرین در بالادست روستای آب‌ماهی تشکیل می‌شود و پس از طی مسیر طولانی حدود ۲۰۰ کیلومتر به دریاچه بختگان می‌ریزد. سد ملاصدرا با مختصات "۵۲°۷'۵۰" شرقی و "۳۰°۳۹'۵۶" شمالی روی رودخانه کر و حدود ۶۰ کیلومتری بالادست سد درودزن احداث شده است. متوسط آورد سالانه رودخانه کر به این سد ۳۹۰ میلیون کیلومتر در سال است [۷]. شکل زیر موقعیت سد ملاصدرا را در استان فارس نشان می‌دهد. در این تحقیق از داده‌های بارش، دما، تبخیر و دبی، مختص به دوره آماری ۱۳۸۶-۱۳۹۱ استفاده شده و مشخصات آماری این پارامترها در جدول ۱ آورده شده است.

شبکه عصبی مصنوعی

شبکه عصبی مصنوعی ایده‌ای است برای پردازش اطلاعات که از سیستم عصبی زیستی الهام گرفته شده است و مانند مغز انسان به پردازش اطلاعات می‌پردازد. این سیستم از عناصر زیادی به نام نورون تشکیل شده است که برای حل یک مسئله با هم هماهنگ عمل می‌کنند. یک نورون کوچک‌ترین واحد پردازشگر اطلاعات است که اساس عملکرد شبکه عصبی را تشکیل می‌دهد. هر شبکه از یک لایه ورودی، یک لایه خروجی و یک یا چند لایه میانی تشکیل شده است. هر لایه توسط وزن‌ها به نورون‌های لایه بعد متصل می‌شوند. هر نورون مقادیر ورودی را در وزن خود ضرب و با بایاس جمع می‌کند و حاصل به تابع محرکه انتقال داده می‌شود تا به این وسیله به نورون‌های لایه بعد انتقال داده شود. این عمل در نورون‌های لایه‌های شبکه انجام شده تا در نهایت خروجی شبکه به دست آید.

1. Artificial Neural Network
2. Susurluk
3. Maharashtra



شکل ۱. موقعیت سد ملاصدرا و رودخانه کر در استان فارس

جدول ۱. مشخصات آماری پارامترهای مطالعه شده در دوره مد نظر

| پارامتر | تعداد داده‌ها | کمینه | بیشینه | میانگین | انحراف از معیار |
|---------|---------------|-------|--------|---------|-----------------|
| دما | ۲۰۶۵ | -۴۹/۵ | ۴۹/۹۵ | ۱۴/۰۵۹۴ | ۸/۶۷۳۶ |
| دبی | ۲۰۶۵ | ۰ | ۱۳/۳۵۷ | ۰/۵۲۳۴ | ۰/۵۳۴ |
| تبخیر | ۲۰۶۵ | ۰ | ۰/۰۹۲ | ۰/۰۰۹۱ | ۰/۰۱۰۱ |
| بارندگی | ۲۰۶۵ | ۰ | ۷۳ | ۰/۸۵۶۲ | ۴/۷۸۳۷ |

توابع یادگیری

توابع یادگیری بررسی شده در این تحقیق ۱۵ تابع `learnis` `learnhd` `learnh` `learnngdm` `learnngd` `learncon` `learnpn` `learnp` `learnos` `learnlv2` `learnlv1` `learnk` `learnsomb` `learnsomb` و `learnwh` است که در زیر به‌طور مختصر توضیحاتی در خصوص این توابع داده شده است [۹-۱۱].

`learncon`: در برخی شبکه‌ها وزن‌های بردار ورودی هرگز آموزش نمی‌بینند، بنابراین برای رفع این مشکل از یک بایاس استفاده می‌شود و برای به‌روزرسانی بایاس‌ها از تابع `learncon` استفاده می‌شود.

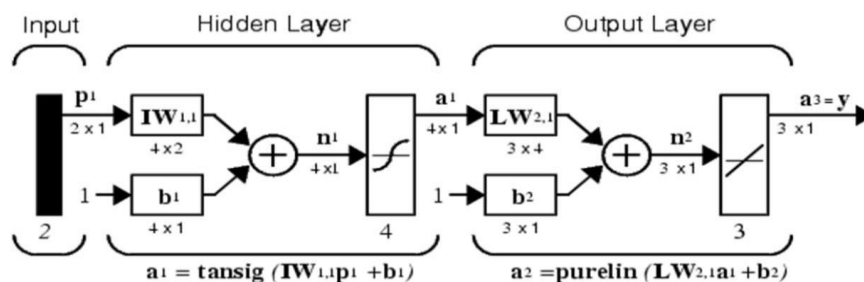
`learnngd`: این تابع از گرادیان نزولی برای تغییر وزن‌ها بهره می‌برد.

`learnngdm`: این تابع از گرادیان نزولی دسته‌ای (دسته‌بندی شده) برای تغییر وزن‌ها بهره می‌برد.

شبکه پرسپترون چندلایه

۹۰ درصد شبکه‌های عصبی مصنوعی که در مسائل مختص به آب استفاده می‌شود از نوع الگوریتم پس‌انتشار خطا^۱ هستند [۸]. در شبکه عصبی پرسپترون چندلایه با الگوریتم پس‌انتشار خطا در لایه پنهان از تابع محرکه تانژانت سیگموئید (`tansig`) و در لایه خروجی از تابع محرکه خطی (`purelin`) استفاده می‌کند و تابع یادگیری وزن‌های تابع (`learnngdm`) است. در این شبکه‌ها اطلاعات ورودی توسط نورون‌های ورودی دریافت و سپس به نورون‌های لایه پنهان فرستاده می‌شود و پردازش مد نظر روی آنها انجام می‌گیرد و نتیجه به نورون‌های لایه خروجی فرستاده شده تا به‌عنوان خروجی شبکه نمایش داده شوند. شکل ۱ نمونه‌ای از یک شبکه پرسپترون چندلایه را نشان می‌دهد.

1. Backpropagation



شکل ۲. ساختار شبکه عصبی پرسپترون چندلایه [۷]

مقادیر ورودی حرکت می‌دهد و این عمل، به ورودی‌ها اجازه می‌دهد تا از بردار تحریک شده استفاده کنند. **learnp**: قاعده یادگیری پرسپترون (learnp) تغییرات مطلوب را برای وزن‌ها و بایاس‌های پرسپترون یا استفاده از ورودی و خطا محاسبه می‌کند. قاعده پرسپترون به همگرا شدن شبکه در تعداد متناهی از تکرارها منجر می‌شود. **learnpn**: از آنجا که در برخی بردارهای ورودی پرسپترون ممکن است مقادیر بسیار بزرگی وجود داشته باشد، این مقادیر بزرگ مشکلاتی را در روال آموزش پرسپترون و در اصلاح وزن‌ها ایجاد می‌کند و در نتیجه تعداد زیادی داده دیگر نیاز است تا با تعدیل این داده‌ها نتیجه مطلوب به دست آید. این تابع تعدیل شده برای اصلاح وزن‌ها است تا مشکل یادشده حل شود.

learnsom: این تابع بهترین نورون‌ها را تشخیص و وزن‌های نورون برگزیده و نورون‌های دیگر را تغییر می‌دهند که در همسایگی آن قرار دارند تا به بردار ورودی نزدیک‌تر شوند. شایان یادآوری است وزن‌های نورون‌های برنده (برگزیده) متناسب با سرعت یادگیری تغییر می‌یابد. **learnsomb**: این تابع تغییر وزن‌های هر نورون، در بردار وزن‌های جدید را محاسبه می‌کند و میانگین وزنی بردارهای ورودی نورون و نورون‌هایی که در همسایگی آن هستند را با خروجی ۱ نشان می‌دهد.

learnwh: الگوریتم یادگیری Widrow-Hoff بر مبنای رویه تقریب بیشترین کاهش^۲ عمل می‌کند.

محاسبه پارامترهای برآورد خطا

مبنای تصمیم‌گیری برای انتخاب بهترین شبکه در هر بار اجرای برنامه، مقدار مجموع مربعات خطا^۳ (SSE)، است و در کنار آن میانگین قدرمطلق خطا^۴ (MAE) و میانگین

learnh: این تابع که به تابع یادگیری Hebb نیز نامیده می‌شود وزن‌های نورون‌های ورودی را نسبت به وزن نورون‌های خروجی افزایش می‌دهد و این عمل اجازه می‌دهد که نورون‌ها ارتباط مؤثرتری بین لایه ورودی و خروجی برقرار کنند.

learnhd: این تابع یادگیری نیز شبیه تابع **learnh** است، اما مؤلفه‌ای به آن اضافه شده است که به صورت نمایی وزن‌ها را در هر گام زمانی کاهش می‌دهد تا بدین صورت مشکل تابع **Hebb** رفع شود که افزایش وزن‌ها بدون وجود هیچ‌گونه محدودیتی بوده است.

learnis: این تابع اجازه می‌دهد تا سلول‌های عصبی، ارتباط بین بردارهای ورودی و خروجی خود را کشف کنند.

learnk: این تابع ابزاری برای اعمال قانون یادگیری کوهن^۱ است. قانون کوهن به وزن‌های یک نورون قابلیت یادگیری می‌دهد و از این منظر در کاربردهای تشخیص الگو مفیدتر است.

learnlv1: یادگیری LVQ بر مبنای مجموعه‌ای از زوج‌های ورودی یا هدف است. در تابع **learnlv1** روی یک ردیف از بردار وزن‌های ورودی و بدون اثر روی دیگر ردیف‌های بردار ورودی وزن‌ها، اصلاحاتی به صورت خودکار انجام می‌شود این عمل با پخش خطای خروجی به لایه اول، انجام می‌پذیرد.

learnlv2: روال یادگیری در این تابع نیز مانند تابع **learnlv1** است با این تفاوت که دو بردار از لایه اول به روزرسانی می‌شوند که یکی از آنها متعلق به کلاس درست و دیگری مختص به کلاس نادرست است.

learnos: این تابع متفاوت از قاعده یادگیری **learnis** عمل می‌کند. این تابع وزن بردارهایی که از لایه ورودی به سمت بردار خروجی می‌آیند را متناسب با اندازه گام‌های

2. Approximate Steepest Descend Procedure
3. Sum Square Error
4. Mean Absolute Error

1. Kohonen Learning Rule

جدول ۲. نتایج آزمون روند من- کندال اصلاح شده

| پارامتر | مقدار آماره z | سطح اطمینان | حد پایین مجاز | حد بالای مجاز |
|---------|---------------|-------------|---------------|---------------|
| دما | -۱/۷۱ | ۹۵ درصد | -۱/۹۶ | ۱/۹۶ |
| دبی | ۰/۶۷۱۸ | ۹۵ درصد | -۱/۹۶ | ۱/۹۶ |
| تبخیر | ۱/۵۶۲ | ۹۵ درصد | -۱/۹۶ | ۱/۹۶ |
| بازندگی | ۰/۱۷۱۸ | ۹۵ درصد | -۱/۹۶ | ۱/۹۶ |

برای مدل سازی هر پدیده، مثلاً میزان دبی جریان رودخانه با هر مدل دیگر باید ابتدا دریافت که چه متغیرهایی روی آن پدیده اثر دارند تا بتوان آن متغیرها را به عنوان ورودی مدل در نظر گرفت. در این تحقیق ورودی مدل ANN داده های بارش، رواناب، دما و تبخیر است و از تعداد ۲۰۶۵ داده موجود ۷۰ درصد (۱۴۴۵ داده) به آموزش شبکه ۱۵ درصد داده ها (۳۱۰ داده) به صحت سنجی و ۱۵ درصد (۳۱۰ داده) برای تست شبکه اختصاص داده شد.

نتایج و بحث

شبکه عصبی یک مرتبه، با یک لایه پنهان (ANN1) و یک مرتبه، با دو لایه پنهان (ANN2) ساختار بندی شده و سپس عملکرد توابع یادگیری شبکه عصبی در هر یک از این دو ساختار بررسی شده است. برای مدل ANN1 تعداد نورون ها از یک تا ۱۰ نورون (۱۰ حالت) و در مدل ANN2 نیز به همین ترتیب برای هر لایه از یک تا ۱۰ نورون (۱۰۰ حالت) بررسی شد و در نهایت بهترین ساختار یک لایه و دو لایه برای هر یک از توابع یادگیری با توجه به مقدار MSE صحت سنجی مشخص شد. هر یک از این دو مدل، ده مرتبه اجرا شد و دو تابعی که در هر مرتبه اجرا، عملکرد مناسب تری نسبت به دیگر توابع داشته اند به همراه مناسب ترین تعداد نورون در لایه پنهان، برای مدل ANN1، در جدول ۳ و برای مدل ANN2 در جدول ۴ آورده شده است. همان طور که در جدول ۳ ملاحظه می شود در مدل ANN1 تابع یادگیری learnh و learnl به ترتیب با MSE برابر ۰/۰۰۰۱۵۸، ۰/۰۰۰۱۸۵ و ۰/۰۰۰۱۸۸، عملکرد مناسب تری نسبت به دیگر توابع یادگیری داشته اند. در مدل ANN2 توابع یادگیری learncon و learnsomb به ترتیب با MSE برابر ۰/۰۰۰۱۵۴، ۰/۰۰۰۱۷۳ و ۰/۰۰۰۱۷۶، عملکرد مناسب تری نسبت به دیگر توابع یادگیری داشته اند. خلاصه عملکرد بهترین توابع یادگیری در خصوص پیش بینی رواناب روزانه در جدول ۵ و شکل ۳ آورده شده است.

مربعات خطا^۱ (MSE) نیز محاسبه شده که روابط این سه معیار عملکرد مطابق روابط ۱-۳ است:

$$SSE = \sum_{i=1}^n (Q_i - \hat{Q}_i)^2 \quad (1)$$

$$MAE = \frac{\sum_{i=1}^n |Q_i - \hat{Q}_i|}{n} \quad (2)$$

$$MSE = \frac{\sum_{i=1}^n (Q_i - \hat{Q}_i)^2}{n} \quad (3)$$

در روابط یاد شده Q_i دبی مشاهده شده، \hat{Q}_i دبی پیش بینی شده و n نیز تعداد کل داده ها است.

نرمال سازی داده ها

وارد کردن داده ها به صورت خام سبب کاهش سرعت و دقت شبکه می شود. به همین منظور برای یکسان کردن ارزش داده ها برای شبکه، عمل نرمال سازی داده ها صورت گرفت. این کار مانع کوچک شدن بیش از حد وزن ها و سبب جلوگیری از اشباع زود هنگام نورون ها می شود [۱۲]. روش های مختلفی برای نرمال کردن داده ها وجود دارد که از جمله آنها می توان به نرمال کردن خطی داده ها بین صفر و یک، بین دو عدد خاص و نرمال سازی آماری داده ها اشاره کرد [۷]. با توجه به اینکه خروجی تابع سیگموئید بین صفر و یک است نرمال سازی داده ها با استفاده از رابطه ۴ بین صفر و یک انجام گرفت.

$$X_n = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \quad (4)$$

که در این رابطه X_n مقدار داده نرمال شده، X_i مقدار داده مشاهده ای، X_{\max} و X_{\min} به ترتیب معرف داده مشاهده ای حداکثر و حداقل هستند. پس از نرمال سازی داده ها، ایستایی آنها با استفاده از آزمون من- کندال اصلاح در سطح احتمال ۹۵ درصد بررسی شد که نتایج نشان داد داده ها روند ندارند. خلاصه نتایج آزمون من- کندال اصلاح شده در جدول ۲ آورده شده است. برای آشنایی با آزمون من- کندال اصلاح شده به کومار و همکارانش (۲۰۰۹) مراجعه شود [۱۳].

جدول ۳. نتایج به دست آمده از مدل ANN1

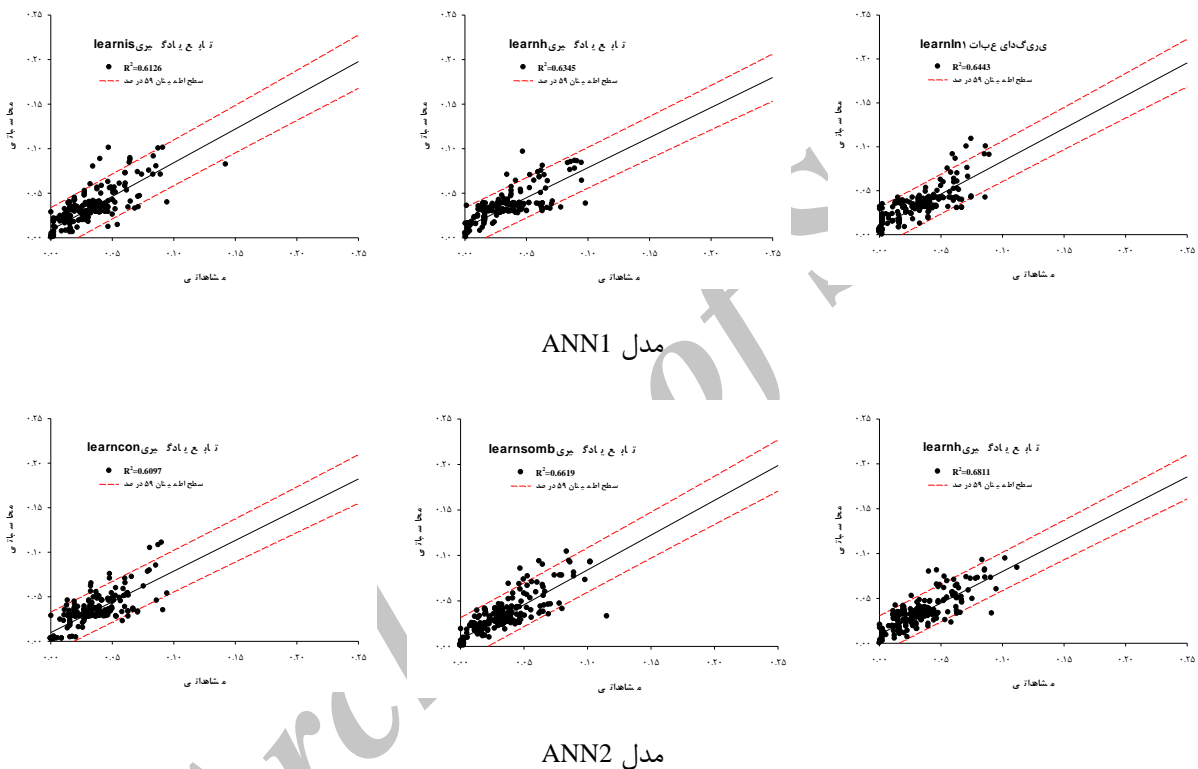
| مرتبه اجرا | تابع یادگیری | تعداد نورون | | عملکرد بهترین توابع یادگیری | | | عملکرد تولباکس شبکه عصبی | | |
|------------|--------------|-------------|----------|-----------------------------|----------|----------|--------------------------|----------|----------|
| | | لایه اول | لایه دوم | MSE | MAE | SSE | MSE | MAE | SSE |
| ۱ | learnhd | ۷ | * | ۰/۰۰۰۲۳۸ | ۰/۰۱۱۰۹۴ | ۰/۰۶۰۵۱۵ | ۰/۰۰۰۵۱۲ | ۰/۰۱۳۲۲۴ | ۰/۱۲۹۹۶۸ |
| | learnwh | ۴ | * | ۰/۰۰۰۲۵۶ | ۰/۰۱۱۷۴۳ | ۰/۰۶۴۹۱۴ | | | |
| ۲ | learnghm | ۸ | * | ۰/۰۰۰۲۱۵ | ۰/۰۱۱۰۳۵ | ۰/۰۵۴۶۵۲ | ۰/۰۰۰۲۱۵ | ۰/۰۱۱۰۳۵ | ۰/۰۵۴۶۵۲ |
| | learnwh | ۷ | * | ۰/۰۰۰۲۲۲ | ۰/۰۱۰۱۷۵ | ۰/۰۶۵۳۸۵ | | | |
| ۳ | learnis | ۸ | * | ۰/۰۰۰۱۸۸ | ۰/۰۰۹۴۴۳ | ۰/۰۴۷۷۴۲ | ۰/۰۰۰۲۵۷ | ۰/۰۱۲۷۹۱ | ۰/۶۵۳۶۱ |
| | learnghm | ۳ | * | ۰/۰۰۰۲۵۷ | ۰/۰۱۲۷۹۱ | ۰/۰۵۵۳۶۱ | | | |
| ۴ | learnlv1 | ۵ | * | ۰/۰۰۰۲۰۸ | ۰/۰۱۰۳۳۶ | ۰/۰۵۲۸۸۱ | ۰/۰۰۰۱۰۱ | ۰/۰۱۴۲۰۶ | ۰/۲۵۶۶۳۶ |
| | learnlv2 | ۵ | * | ۰/۰۰۰۲۳۴ | ۰/۰۱۱۴۵۳ | ۰/۰۵۹۴۷۹ | | | |
| ۵ | learncon | ۶ | * | ۰/۰۰۰۲۳۱ | ۰/۰۱۰۴۴۵ | ۰/۰۵۸۶۰۶ | ۰/۰۰۵۵۵۶ | ۰/۰۲۰۲۴۲ | ۱/۴۱۱۱۳ |
| | learnsomb | ۲ | * | ۰/۰۰۰۲۴۷ | ۰/۰۱۱۷۳۹ | ۰/۰۶۲۶۸۳ | | | |
| ۶ | learnh | ۳ | * | ۰/۰۰۰۱۸۵ | ۰/۰۰۹۸۶۱ | ۰/۰۴۶۸۸۱ | ۰/۰۰۰۳۸۷ | ۰/۰۱۳۰۴۵ | ۰/۰۹۸۲۶ |
| | learnsomb | ۵ | * | ۰/۰۰۰۲۵۷ | ۰/۰۱۱۷۰۹ | ۰/۰۶۵۱۹۸ | | | |
| ۷ | learnlv1 | ۴ | * | ۰/۰۰۰۱۵۸ | ۰/۰۰۹۰۷۹ | ۰/۰۴۰۲۳۲ | ۰/۰۰۰۲۴۱ | ۰/۰۱۱۰۲۴ | ۰/۰۶۱۲۵۶ |
| | learncon | ۱۰ | * | ۰/۰۰۰۲۰۵ | ۰/۰۰۹۸۱۵ | ۰/۰۵۲۰۳۶ | | | |
| ۸ | learnpn | ۷ | * | ۰/۰۰۰۲۲۵ | ۰/۰۱۰۳۹۶ | ۰/۰۵۷۱۴۶ | ۰/۰۰۱۰۸ | ۰/۰۱۲۸۱۳ | ۰/۲۷۴۳۰۱ |
| | learnhd | ۸ | * | ۰/۰۰۰۲۵۶ | ۰/۰۱۰۵۴۳ | ۰/۰۶۴۹۴۱ | | | |
| ۹ | learnsomb | ۳ | * | ۰/۰۰۰۲۴۵ | ۰/۰۱۱۱۴۱ | ۰/۰۶۲۲۲۷ | ۰/۰۰۰۳۰۱ | ۰/۰۱۱۵۶۴ | ۰/۰۷۶۵۲۴ |
| | learnghm | ۸ | * | ۰/۰۰۰۳۰۱ | ۰/۰۱۱۵۶۴ | ۰/۰۷۶۵۲۴ | | | |
| ۱۰ | learnsomb | ۶ | * | ۰/۰۰۰۲۴۵ | ۰/۰۱۱۴۸۱ | ۰/۰۶۲۲۵۲ | ۰/۰۰۴۱۰۷ | ۰/۰۱۷۲۳۷ | ۱/۰۴۳۲۹۹ |
| | learngh | ۲ | * | ۰/۰۰۰۲۸۴ | ۰/۰۱۲۶۹۸ | ۰/۰۷۲۰۸۳ | | | |

جدول ۴. نتایج به دست آمده از مدل ANN2

| مرتبه اجرا | تابع یادگیری | تعداد نورون | | عملکرد بهترین توابع یادگیری | | | عملکرد تولباکس شبکه عصبی | | |
|------------|--------------|-------------|----------|-----------------------------|----------|----------|--------------------------|----------|-----------|
| | | لایه اول | لایه دوم | MSE | MAE | SSE | MSE | MAE | SSE |
| ۱ | learnlv2 | ۳ | ۲ | ۰/۰۰۰۱۹۲ | ۰/۰۰۹۵۲۴ | ۰/۰۴۸۶۷۹ | ۰/۰۰۰۲۰۸ | ۰/۰۱۰۹۲۴ | ۰/۰۵۲۹۴۷ |
| | learnghm | ۶ | ۸ | ۰/۰۰۰۲۰۸ | ۰/۰۱۰۹۲۴ | ۰/۰۵۲۹۴۷ | | | |
| ۲ | learnh | ۱۰ | ۳ | ۰/۰۰۰۱۵۴ | ۰/۰۰۹۱۱۸ | ۰/۰۳۹۲۳۸ | ۰/۰۰۰۵۵۹ | ۰/۰۱۱۷۹۵ | ۰/۰۱۴۱۹۶۲ |
| | learnsomb | ۵ | ۱۰ | ۰/۰۰۰۱۷۳ | ۰/۰۰۹۱۳۹ | ۰/۰۴۳۹۰۷ | | | |
| ۳ | learnos | ۸ | ۹ | ۰/۰۰۰۱۸۵ | ۰/۰۰۹۸۸۴ | ۰/۰۴۷۰۴۶ | ۰/۰۰۳۱۳۷ | ۰/۰۱۳۹۶۲ | ۰/۷۹۶۹۰۶ |
| | learnh | ۱۰ | ۱۰ | ۰/۰۰۰۱۹۹ | ۰/۰۱۰۶۲۸ | ۰/۰۵۰۶۰۹ | | | |
| ۴ | learnos | ۳ | ۶ | ۰/۰۰۰۱۷۷ | ۰/۰۰۸۹۶۴ | ۰/۰۴۴۹۵۱ | ۰/۰۰۳۰۹۴ | ۰/۰۱۴۵۵۲ | ۰/۷۸۵۸۲۹ |
| | learngh | ۸ | ۹ | ۰/۰۰۰۲۳ | ۰/۰۱۰۶۲۷ | ۰/۰۵۸۵۰۹ | | | |
| ۵ | learnos | ۵ | ۲ | ۰/۰۰۰۲۳۶ | ۰/۰۱۰۳۸۱ | ۰/۰۵۹۹۳ | ۰/۰۰۴۰۸ | ۰/۰۱۵۵۶۷ | ۱/۰۳۶۲۷۸ |
| | learnk | ۸ | ۳ | ۰/۰۰۰۲۳۹ | ۰/۰۱۰۸۵۳ | ۰/۰۶۰۷۵۲ | | | |
| ۶ | learncon | ۶ | ۳ | ۰/۰۰۰۱۷۶ | ۰/۰۰۹۵ | ۰/۰۴۴۵۹۳ | ۰/۰۰۰۲۰۵ | ۰/۰۰۹۷۹۸ | ۰/۰۵۲۰۲۲ |
| | learnh | ۷ | ۲ | ۰/۰۰۰۱۹۵ | ۰/۰۰۸۹۸۸ | ۰/۰۴۹۶۲ | | | |
| ۷ | learnlv1 | ۸ | ۶ | ۰/۰۰۰۱۸۸ | ۰/۰۰۹۱۶۶ | ۰/۰۴۷۸۶۶ | ۰/۰۰۱۶۱۹ | ۰/۰۱۲۸۰۱ | ۰/۴۱۱۲۲۲ |
| | learnis | ۱۰ | ۷ | ۰/۰۰۰۲۲ | ۰/۰۱۰۰۸۲ | ۰/۰۵۵۸۸۹ | | | |
| ۸ | learnos | ۶ | ۹ | ۰/۰۰۰۱۹۳ | ۰/۰۱۰۵۴۴ | ۰/۰۴۸۹۳ | ۰/۰۰۳۴۱۸ | ۰/۰۱۵۱۸۵ | ۰/۸۶۸۱۱۵ |
| | learnlv1 | ۷ | ۷ | ۰/۰۰۰۲۰۲ | ۰/۰۱۰۰۵۱ | ۰/۰۵۱۴۰۵ | | | |
| ۹ | learnwh | ۱۰ | ۶ | ۰/۰۰۰۱۹۷ | ۰/۰۰۹۸۱۵ | ۰/۰۴۹۹۱۳ | ۰/۰۰۰۲۵۱ | ۰/۰۱۰۵۵۶ | ۰/۰۶۳۷۵۵ |
| | learnh | ۳ | ۷ | ۰/۰۰۰۲ | ۰/۰۱۰۰۳۱ | ۰/۰۵۰۸۴۱ | | | |
| ۱۰ | learnp | ۴ | ۹ | ۰/۰۰۰۲۰۳ | ۰/۰۱۰۲۶۴ | ۰/۰۵۱۵۹۴ | ۰/۰۰۰۸۴۵ | ۰/۰۰۹۹۱۵ | ۰/۲۱۴۶۱۷ |
| | learnpn | ۸ | ۴ | ۰/۰۰۰۲۱۲ | ۰/۰۰۹۱۷۱ | ۰/۰۵۳۸۵۳ | | | |

جدول ۵. خلاصه عملکرد بهترین توابع یادگیری

| تابع | MSE | R | |
|-----------|----------|-------|------|
| learnlv1 | ۰/۰۰۰۱۵۸ | ۰/۸۰۰ | ANN1 |
| learnh | ۰/۰۰۰۱۸۵ | ۰/۷۹۷ | |
| learnis | ۰/۰۰۰۱۸۸ | ۰/۷۸۳ | |
| learnh | ۰/۰۰۰۱۵۴ | ۰/۸۲۵ | ANN2 |
| learnsomb | ۰/۰۰۰۱۷۳ | ۰/۸۱۴ | |
| learncon | ۰/۰۰۰۱۷۶ | ۰/۷۸۱ | |



شکل ۳. نتایج بررسی همبستگی بین داده‌های مشاهداتی و محاسباتی

بیشتر است. همچنین در جدول ۳ ملاحظه می‌شود که شبکه عصبی با دو لایه پنهان همواره بهتر از شبکه با یک لایه پنهان است، اما تفاوت آنها چشمگیر نیست تحقیقات تئوریک انجام شده نیز بیانگر آن است که یک لایه پنهان نیز می‌تواند کارایی لازم را داشته باشد [۱۴-۱۶]. همچنین تحقیقات تجربی و عملی این موضوع را تأیید می‌کنند [۱۶ و ۱۷].

۹۰ درصد شبکه‌های عصبی مصنوعی که در مسائل هیدرولوژیکی استفاده می‌شوند از نوع الگوریتم پس‌انتشار خطا (newff) هستند که در مسائل پیش‌بینی سری زمانی

نتایج نشان می‌دهند در ANN1 در هر مرتبه اجرا، توابع یادگیری، عملکرد متفاوتی داشته‌اند؛ اما دو تابع یادگیری learnsgdm و learnsom به ترتیب با چهار و سه مرتبه تکرار در بین بهترین توابع یادگیری عملکرد مناسب‌تری داشته‌اند. همچنین در مدل ANN2، نیز تابع یادگیری learnh و learnos با چهار مرتبه تکرار در بین بهترین توابع یادگیری، عملکرد مناسب‌تری داشته است. بنابراین، با توجه به این موضوع می‌توان گفت که اگر از توابع یاد شده برای یادگیری وزن‌های ورودی و خروجی شبکه عصبی استفاده شود شانس رسیدن به جواب مناسب

منابع

- [1]. Tokar AS, Markus M. Precipitation – runoff modeling using artificial neural network and conceptual models. *Journal of Hydrologic Engineering*. 2000;4:150-161.
- [2]. Razavi SS, Karamuoz M. in Prediction monthly river flows by using artificial neural network. 10th students Conference on Civil Engineering. Amirkabir University of Technology. 22 Oct 2003. [Persian]
- [3]. Fathi P, Mohammadi Y, Homayi M. Intelligent modeling of monthly flow time series into vahdat dam in sanandaj city. *Journal of Water and Soil*. 2009; 23(1):209-220. [Persian]
- [4]. Dorum A, Yazar A, FaikSevimli M, and Onucyildiz M. Modelling the rainfall-runoff data of susurluk basin. *Expert Systems with Applications*. 2010. 37: 6587-6593.
- [5]. Chua HC, and Wong SW. Rainfall-runoff modeling using artificial neural network coupled with singular spectrum analysis. *Journal Hydrology*. 2011. 399: 394-409.
- [6]. Patil S, Valunjkar, S. Study of different rainfall-runoff forecasting algorithms for better water consumption. *International Conference on Computational Techniques and Artificial Intelligence*. 2012. 327-330.
- [7]. Zeynali MJ, Nikbakht S, Mohammadezapour O. Prediction Input Flows to Mollasadra Reservoir by Using Artificial Neural Network. 5th Iranian water resources management conference. Shahidbehshiti University. 29 Jul 2013. [Persian]
- [8]. Braddock RD, Kremmer ML, Sanzogni L. Feedforward artificial neural network model for forecasting rainfall-runoff. *Journal of Environmental Sciences*. 1998. 9:419-432.
- [9]. Kia M. *Soft Computing in MATLAB*. Qian academic publishing. [Persian]
- [10]. www.mathwork.com
- [11]. Demuth H, Beale M. *Neural network toolbox for use with MATLAB*. Sixth printing Revised for Version 4. Pp:680.
- [12]. Hahangeer AR, Raeini M, Ahmadi MZ. Comparison of artificial neural networks (ANN) simulation of rainfall-runoff process with HEC-HMS model in Kardeh watershed. *Journal of Water and Soil*. 2008. 22(2):72-84. [Persian]
- [13]. Kumar S, Merwade V, Kam J, Thurner K. Streamflow trends in Indiana: effects of long term persistence, precipitation and subsurface drains. *Journal of Hydrology*. 2009. 374(1): 171-183.

نیز از این الگوریتم استفاده می‌شود. در شبکه پس انتشار خطا (newff) به صورت پیش فرض، تابع انتقال در لایه پنهان، tansig، تابع انتقال در لایه خروجی تابع purelin و تابع یادگیری پیش فرض شبکه تابع learnnrm است. در این تحقیق نیز کلیه پیش فرض های شبکه پذیرفته شده است و تنها تابع یادگیری تغییر کرده اند و نتایج نشان می‌دهد تابع learnnrm برای ساختار شبکه با یک لایه پنهان مناسب است و در بیشتر تحقیقات نیز فقط از یک لایه پنهان استفاده می‌شود و فقط آنالیز حساسیت روی تعداد نورون ها انجام می‌گیرد [۱۸ و ۱۹]

نتیجه گیری

در این تحقیق مدل شبکه عصبی با ورودی های بارش، دما، تبخیر و رواناب و در دو ساختار یک و دو لایه پنهان بررسی شد که به ترتیب ANN1 و ANN2 نامگذاری شدند و عملکرد ۱۵ تابع یادگیری بررسی شد و نتایج نشان با توجه به معیار MSE در مدل ANN1 در هر مرتبه اجرا توابع عملکرد متفاوتی دارند. اما توابع learnnrm، learnnrm و learnnrm مدل ANN2 توابع یادگیری learnnrm، learnnrm و learnnrm عملکرد مناسب تری نسبت به دیگر توابع یادگیری داشته اند. همچنین تابع learnnrm که تابع پیش فرض شبکه عصبی است در مدل ANN1 عملکرد بهتری دارد و بنابراین هنگام استفاده از تولباکس شبکه عصبی بهتر است تعداد لایه پنهان بیشتر از یک نباشد؛ زیرا در این صورت شانس رسیدن به جواب مناسب بیشتر خواهد بود.

از یک سو، توابعی مانند learnnrm و learnnrm مناسبی داشته اند؛ اما از سوی دیگر دو تابع یادگیری learnnrm و learnnrm در مدل ANN1 و learnnrm در مدل ANN2، بیشترین تکرار را در بین بهترین توابع یادگیری، داشته اند و با توجه به این موضوع که تعداد دفعاتی که یک تابع در بین توابع با عملکرد مناسب، قرار می‌گیرد به مراتب مهم تر از بهترین بودن است؛ بنابراین برای ساختارهای شبکه عصبی با یک لایه می‌توان از learnnrm کنار learnnrm استفاده کرد و اگر تعداد لایه پنهان بیشتر باشد، باید با احتیاط از تولباکس شبکه عصبی (ntstool) و تابع newff استفاده کرد.

- [14].Cybenko G. Approximation by superposition of a sigmoidal function. *Mathematics of control, signals and systems* 2.4. 1989. 303-314.
- [15].Hornik K, Stinchcombe M, White H. Multilayer feed-forward networks are universal approximators. *Neural Networks*. 1989. 2(5):359-366.
- [16].Zhang G, Patuwo BE, Hu MY. Forecasting with artificial neural networks: the state of the art. *International Journal of Forecasting*. 1998. 14(1):35-62.
- [17].Noori R, Abdoli MA, Ghasrodashti AA, JaliliGhazizade M. Prediction of municipal solid waste generation with combination of support vector machine and principal component analysis: a case study of Mashhad. *Environmental Progress & Sustainable Energy*. 2009. 28 (2):249-258.
- [18].Nikmanesh MR. Prediction of Monthly Average Discharge Using the Hybrid Model of Artificial Neural Network and Wavelet Transforms (Case Study: Kor River Pol-e-Khan Station). *Journal of Water and Soil Conservation*. 2015. 22(3):231-239. [Persian]
- [19].Noori R, Farokhnia A, Morid S, RiahiMadvar H. Effect of Input Variables Preprocessing in Artificial Neural Network on Monthly Flow Prediction by PCA and Wavelet Transformation. *Journal of Water & Wastewater*. 2008. 20(69):1-22.

Archive of SID