Archive of SID

تخمین تلاش لازم برای ساخت سیستم اطلاعاتی با استفاده از شبکهٔ عصبی RBF

سرهنگ دکتر غلامرضا شاه محمدی[1]

دریافت مقـاله: ۸۹/۰۳/۰۹
پـذیرش مقاله: ۸۹/۱۱/۲۸

چکیده

یکی از ابعاد مهم مدیریت پروژه‌های نرم‌افزاری تخمین هزینه و مدت‌زمان موردنیاز برای ساخت سیستم اطلاعاتی است. تخمین یا برآورد تلاش موردنیاز برای ساخت یک سیستم اطلاعاتی یکی از دغدغه‌های مهم مدیر پروژه، تلقی می‌شود. در نتیجه، الگوهای تخمین تلاش متعددی مطرح شده است. روش‌های مبتنی بر یادگیری، مانند شبکه‌های عصبی، یکی از این الگوها می‌باشد. هدف از طرح این تحقیق، استفاده از شبکهٔ عصبی RBF[2] برای تخمین تلاش لازم برای ساخت سیستم اطلاعاتی می‌باشد. نتایج حاصل از این بررسی نشان می‌دهد که این شبکه در مقایسه با روش‌های مبتنی بر الگو، تخمین مناسبی از تلاش‌های لازم برای ساخت سیستم را ارائه داده است.

کلیدواژه:
شبکهٔ عصبی Neural Network، سیستم اطلاعاتی Information System، تخمین تلاش Cost Estimation،

---

۱. عضو هیأت علمی دانشگاه علوم انتظامی

2. Radial Basis Function

www.SID.ir

Archive of SID

۱. مقدمه

منظور از تخمین تلاش، پیش‌بینی مقدار تلاش، زمان و تعداد کارکنان موردنیاز برای ساخت سیستم‌های اطلاعاتی می‌باشد. اجزای اصلی هزینهٔ پروژه، هزینه‌های سخت‌افزاری، آموزش و هزینه‌های مربوط به تولید و ساخت نرم‌افزار، شامل: پرداخت دستمزد به مهندسان نرم‌افزار است. از آن‌جاکه، بخش اعظم هزینهٔ پروژه، هزینهٔ کارکنان است؛ اصطلاح تخمین هزینه و تخمین تلاش به‌صورت متقابل استفاده می‌شوند. تخمین در ابتدای فرایند ساخت سیستم، که به تخمین مقدماتی معروف است، اغلب دقت کمی دارد؛ زیرا در ابتدا، دانش کمی از پروژه موجود است.

مدیریت موفق پروژه نقش مؤثری در بهره‌وری سازمان، ایفا می‌نماید. بنابراین، استفاده از روش‌های مؤثر تخمین هزینه و تلاش، تأثیر قابل‌توجهی را در مدیریت پروژه دارد. درصورت وجود اطلاعات، تلاش و هزینهٔ سیستم‌های اطلاعاتی تولید شده، بهره‌برداری از شیوهٔ پیشنهادی در این مقاله، نقش مؤثری در تخمین تلاش و هزینهٔ سیستم‌های اطلاعاتی، که در سازمان‌ها تولید خواهند شد، دارند.

تخمین تلاش موردنیاز برای ساخت یک سیستم اطلاعاتی، یکی از دغدغه‌های مهم مدیریت پروژه، تلقی می‌شود. الگوهای تخمین هزینه‌ای که در مراحل اولیه ساخت پروژه، با حداقل اطلاعات موجود از پروژه، هزینهٔ ساخت سیستم را تخمین می‌زنند، سودمند و موردنیاز هستند. روش تخمین هزینهٔ مناسب، امکان کنترل مؤثر زمان و هزینهٔ ساخت سیستم را فراهم می‌نماید (کیم و همکاران، ۲۰۰۴). در نتیجه الگوهای متعددی برای تخمین تلاش سیستم مطرح شده است.

در بررسی بوهم و همکارانش، الگوهای تخمین تلاش به چهار دسته تقسیم شده‌اند؛ شامل روش‌های مبتنی بر: الگو، نظر خبره، یادگیری و پویایی هستند. روش‌های مبتنی بر الگو، از متداول‌ترین شیوه تخمین تلاش سیستم می‌باشند (باری، آبتس و کولانی، ۲۰۰۰).

بوهم (۱۹۸۱)، معتقد است که تخمین تلاش ساخت سیستم در مراحل اولیه ساخت سیستم بین ۲۵ تا ۴۰۰ درصد تلاش واقعی، متغیر است؛ به‌عبارت دیگر، تخمین تلاش از دقت بسیار کمی برخوردار است؛ زیرا دانش تخمین زننده از پروژه

www.SID.ir

Archive of SID

اندک است. این نظر توسط همسترا (۱۹۹۲) نیز تأیید شده است.

شبکهٔ عصبی مصنوعی[1] یا ANN معمول‌ترین روش مبتنی بر یادگیری است که از آن برای تخمین تلاش توسعهٔ نرم‌افزارها استفاده می‌شود (گری و مک‌دونل، ۱۹۹۷). به‌طور نسبی، کاربرد شبکهٔ عصبی در تخمین تلاش در یک پژوهش، جدید است. در سال‌های اخیر، تحقیقات متعددی برای استفاده از روش‌های غیرالگوریتمی نظیر شبکهٔ عصبی مصنوعی، به‌عنوان جایگزین روش‌های مبتنی بر الگو، انجام شده است (ادری و همکاران، ۲۰۰۲). در دو دههٔ گذشته از شبکهٔ عصبی برای پیش‌بینی در کاربردهای مختلفی استفاده و نتایج حاصله در مقایسه با روش‌های معمول بهتر بوده است (باربرو، ۱۹۹۶؛ بری و لینوف، ۱۹۹۷؛ بیگاس، ۱۹۹۶). روش‌های مبتنی بر الگو، در اوایل پروژه، به‌دلیل دانش کمی از پروژه، تخمین مناسبی ارائه نمی‌دهند؛ درحالی‌که، روش‌های مبتنی بر شبکهٔ عصبی مصنوعی، می‌توانند درصورت وجود اطلاعات از تلاش پروژه‌های کامل شده، تخمین تلاش مناسبی را ارائه دهند.

هدف از این بررسی استفاده از شبکهٔ عصبی برای تخمین تلاش سیستم است. ساختار مقاله به این‌صورت است که در بخش دوم روش‌های تخمین تلاش مبتنی بر الگو، معرفی می‌شوند. در بخش سوم شبکه‌های عصبی مصنوعی و شبکهٔ عصبی RBF به‌طور اجمالی معرفی می‌شوند. در بخش چهارم ارزیابی کارایی الگوهای تخمین تلاش و در بخش پنجم، تخمین تلاش با شبکهٔ عصبی RBF ارائه می‌گردند.

۲. روش‌های تخمین تلاش مبتنی بر الگو

روش‌های مبتنی بر الگو، متداول‌ترین برآوردکننده تلاش ساخت نرم‌افزار هستند. الگوی COCOMO[2] و نقاط وظیفه‌ای[3] یا به اختصار FP مثال‌هایی از روش‌های مبتنی بر الگو می‌باشند. COCOMO مبتنی برتحلیل رگرسیون داده‌های مربوط به ۶۳ پروژه کامل‌شده است. الگوی اساسی COCOMO به‌صورت رابطه شمارهٔ (۱)

1. Artificial Neural Network
2. Constructive Cost Model
3. Function Points

www.SID.ir

Archive of SID

نشان داده شده است. در این رابطه، EF تعداد نفر ـ ماه یا ساعات موردنیاز، c مقدار ثابت به اندازهٔ تخمینی ۳/۲، LOC تعداد خطوط برنامه به هزار و k، مقدار ثابت به اندازهٔ تخمینی ۱/۰۵ می‌باشند.

$$EF= c(LOC)^k \quad (1)$$

در الگوی COCOMO تخمین تلاش، شامل سه مرحله است: مرحلهٔ اول، مقدار LOC موردنیاز سیستم اطلاعاتی محاسبه می‌شود. سپس، تلاش کلی با استفاده از معادلهٔ $EF=3/2(LOC)^{1/05}$ تخمین زده می‌شود. مرحلهٔ دوم، تلاش تخمین‌زده شده با استفاده از محرک‌های هزینه[1] یا تنظیم‌کننده‌های تلاش دقیق می‌شود. روش COCOMO دارای سه الگو است که در مراحل مختلف پروژه مورد استفاده قرار می‌گیرند. برای مثال، در الگوی COCOMO میانی، تلاش به‌صورت تابعی از تعداد خطوط برنامه و ۱۵ تنظیم‌کننده تلاش، تخمین زده می‌شود (بوهم، ۱۹۹۵).

روش FP، یک روش جایگزین برای تخمین تلاش توسعه سیستم است که توسط Albrecht مطرح شده است. وی مفهوم FP را برای سنجش نیازهای سیستم مطرح کرده است که در آن ابتدا اندازه سیستم با شناسایی انواع توابع موردنیاز برحسب ورودی‌ها، خروجی‌ها، پرس‌وجوها، فایل‌های داخلی و فایل‌های واسط خارجی تعیین می‌شود. برای محاسبه مقدار FP برای هر دسته، ابتدا تعداد توابع هر دسته در وزن مناسب برحسب پیچیدگی آن ضرب می‌شود. آنگاه تلاش کل سیستم با ضرب FP همه دسته‌ها در عامل پیچیدگی فنی[2] (TCF)، محاسبه می‌شود. TCF با انتساب مقادیر به ۱۴ عامل تأثیرگذار روی پروژه و محاسبه مجموع آن به‌دست می‌آید.

۳. شبکه‌های عصبی مصنوعی

شبکهٔ عصبی مصنوعی، معمول‌ترین روش مبتنی بر یادگیری مورد استفاده برای تخمین تلاش توسعه نرم‌افزار است. شبکهٔ عصبی از عناصر یا واحدهای پردازشی

1. Cost Drivers
2. Technical Complexity Factor

www.SID.ir

Archive of SID

ساده که به‌صورت موازی عمل می‌نمایند، تشکیل شده است (رمود و بیل، ۲۰۰۲). روش سازمان‌دهی نورون‌ها در شبکه، توپولوژی شبکه نامیده می‌شود. توانایی محاسباتی یک نورون محدود است. اما از اتصال آن‌ها به‌یکدیگر سیستم پیچیده‌ای تشکیل می‌شود، که با آموزش قادر به ارائه جواب‌های قابل‌قبول است. برای آموزش شبکهٔ عصبی دو روش اصلی، «یادگیری با نظارت»[1] و «یادگیری بدون نظارت»[2] وجود دارد. در یادگیری بدون نظارت، شبکه به اطلاعات اولیه مربوط به دسته‌بندی درست داده‌های ارائه‌شده نیاز ندارد و می‌تواند مجموعه داده‌های چند بعدی را برای کشف خوشه‌ها[3] و زیرخوشه‌های موجود در داده‌ها، تجزیه و تحلیل نماید. یادگیری با نظارت، مشابه روش یادگیری انسان است؛ به این ترتیب که به شبکه یک سری مثال نشان داده می‌شود.

۳-۱. شبکهٔ عصبی RBF[4]

شبکهٔ RBF، یک شبکه سه لایه است؛ که لایهٔ ورودی یک بافر است. لایه دوم یا لایهٔ مخفی، نگاشتی غیرخطی از فضای ورودی به فضایی با ابعاد بزرگ‌تر است که امکان جداسازی خطی الگوها را میسر می‌سازد. لایهٔ خروجی، مجموع وزنی ساده با خروجی خطی است. اگرهدف از RBF تقریب تابع باشد، این خروجی مناسب است. اما اگر هدف دسته‌بندی الگوها باشد، باید از تابع سیگموید استفاده نمود. ویژگی منحصربه فرد شبکه RBF، فرایند انجام‌شده در لایهٔ مخفی است. اگر الگوها در فضای ورودی، خوشه‌ها را تشکیل دهند و اگر مرکز این خوشه‌ها شناخته شده باشد، فاصله تا مرکز خوشه قابل محاسبه است. سنجش مسافت به‌صورت غیرخطی انجام می‌شود. بنابراین، اگر الگو در ناحیه نزدیک مرکز خوشه باشد، مقدار آن نزدیک به یک خواهد بود. اطراف این ناحیه، مقدار مذکور به‌شدت کاهش می‌یابد. این ناحیه

1. Supervised learning
2. Unsupervised learning
3. Clusters
4. Radial Basis Function

www.SID.ir

Archive of SID

اطراف مرکز خوشه، متقارن است، به‌گونه‌ای که تابع غیرخطی به شعاعی ـ بنیاد[1] شناخته شده که معمول‌ترین تابع آن، تابع گوسین است. مسافت از مرکز خوشه‌ها به‌صورت مسافت اقلیدسی محاسبه می‌گردد. وزن‌های ورودی به هر نورون در لایهٔ مخفی، مختصات مرکز خوشه است. بنابراین، زمانی‌که یک نورون، الگوی ورودی نظیر X را دریافت می‌کند، مسافت از مرکز خوشه از معادله شمارهٔ ۲ به‌دست می‌آید.

$$r_j = \sqrt{\sum_{i=1}^{n} (x_i - w_{ij})^2} \qquad (۲)$$

این شبکه در مقایسه با شبکهٔ عصبی MLP[2] به نورون بیشتری نیاز دارد؛ اما طراحی آن سریع‌تر از شبکهٔ MLP بوده و سریع‌تر آموزش می‌بینند. با تعداد نورون کافی می‌توان هر تابع منطقی را به‌طور تقریبی نشان داد. این شبکه متنوع بوده و دارای توابع مختلف که برخی از آن‌ها به‌شرح ذیل است:

تابع newrbe: شبکه‌ای با خطای صفر برای داده آموزشی ارائه می‌گردد؛

تابع newrb: در یک فرایند تکراری شبکه RBF که به تعداد نورون‌های بیشتری در هر مرحله پرداخته تا جایی‌که خطا از مقدار مورد نظر کمتر شود؛

تابع newgrnn: یک نوع شبکه RBF است، که برای ایجاد توابع تقریبی استفاده می‌شود.

طراحی با شبکه RBF به سهولت انجام می‌گیرد که علاوه بر تعیین تعداد ورودی‌ها و خروجی‌های شبکه، به تعیین عامل Spread که تعیین‌کننده محدودهٔ انتخاب تابع گوسین است، نیاز دارد. بسته نرم‌افزاری MATLAB به‌عنوان یک ابزار مناسب برای کار با شبکه‌های عصبی، دارای انواع متنوع و مختلف شبکه RBF را دارد، که در این بررسی نیز از آن استفاده شده است.

1. Radial-Basis

2. Multilayer Perseptron

www.SID.ir

Archive of SID

۴. ارزیابی کارایی الگوهای تخمین تلاش

دقت تخمین تلاش یک عامل مهم در موفقیت پروژه است (کیم و همکاران، ۲۰۰۴). یک عامل اصلی در انتخاب الگوی تخمینِ تلاش، دقت پیش‌بینی آن در امر تخمین است. به این منظور معیار درصد میانگین مطلق خطا[1] برای ارزیابی دقت تخمین مطرح شده است. مقدار خطا از رابطه شمارهٔ ۳ محاسبه می‌گردد. در این رابطه V_{est} مقدار تلاش تخمین‌زده شده، V_{act}مقدار واقعی تلاش و N تعداد نمونه‌های استفاده‌شده در آزمون است. بررسی Kemerer روی الگوهای FP، COCOMO SLIM، و Esimacs خطای الگوهای مختلف بالا و مقدار خطای مشاهده‌شده بین ۵۷ تا ۸۰۰ بوده است (کمرر، ۱۹۸۷). در بررسی سه الگو توسط Ferens و همکارش با استفاده از پایگاه داده Albrecht و ۱۴ پروژه از مجموعه داده Kemerer معیار بین ۴۶ تا ۱۰۵ درصد گزارش شده است (فرنسی و گومر، ۱۹۹۲). در روش‌های مبتنی بر نظر خبره معیار بین ۳۲ تا ۱۱۰۷ درصد گزارش شده است (ویسی‌نان‌زا و موخوپدهی، ۱۹۹۱).

$$MAPE=(\sum(|V_{est} - V_{act}|/ V_{act})/(N*۱۰۰) \qquad (۳)$$

۵. تخمین تلاش با شبکهٔ عصبی RBF

همان‌گونه که در تخمین تلاش مبتنی بر الگو اشاره شد، تخمین تلاش در دو مرحله انجام می‌گیرد. ابتدا LOC یا FP سیستم تعیین می‌شود. آنگاه سطح پیچیدگی تنظیم‌کننده‌های تلاش تعیین و ضمن انجام محاسبات مربوطه، مقدار تلاش موردنیاز برای ساخت سیستم تعیین می‌گردد. هدف این است که فرایند مرحلهٔ دوم توسط شبکهٔ عصبی انجام گردد؛ یعنی به ازای ورودی LOC یا FP سیستم، شبکهٔ عصبی مقدار تلاش سیستم را تعیین نماید. به این منظور، باید با استفاده از داده‌های مربوط به LOC یا FP و تلاش متناظر با آن، شبکه را آموزش داد. به دلیل عدم وجود

1. Mean Absolute Percentage Error (MAPE)

www.SID.ir

Archive of SID

داده‌های مربوط به LOC یا FP و تلاش متناظر با آن برای پروژه‌های داخل کشور، از داده‌های موجود در مقاله‌های (بیگاس، ۱۹۹۶؛ من‌زیر و همکاران، ۲۰۰۵) استفاده شده است. مجموعه داده مورد استفاده در این بررسی به شرح زیر است:

۱. مجموعه داده ۲۴ پروژه IBM DP Services؛

۲. مجموعه داده ۱۵ پروژه Kemerer؛

۳. مجموعه داده ۲۸ پروژه Halamark؛

۴. مجموعه داده ۶۰ پروژه ناسا.

اندازهٔ پروژه‌ها در مجموعه دادهٔ ۱و۲ متوسط تا بزرگ و در مجموعه داده ۳ متوسط بوده‌اند. مجموعه داده ۴، شامل پروژه‌هایی با اندازهٔ متوسط تا بزرگ است. ورودی شبکه برای مجموعه داده ۱ تا ۳، FP یا LOC (برحسب ۱۰۰۰خط برنامه) و خروجی نفر ـ ساعت تلاش به ۱۰۰۰ است. در مجموعه داده ۴، خروجی به تعداد نفر ـ ماه پروژه است.

جدول شمارهٔ ۱. مجموعه داده IBM DP Service (هیت، ۲۰۰۲)

تعداد خطوط کد به ۱۰۰۰	FPمقدار	تلاش واقعی (برحسب ۱۰۰۰ ساعت)	تعداد خطوط کد به ۱۰۰۰	FPمقدار	تلاش واقعی (برحسب ۱۰۰۰ ساعت)
۱۳۰	۱۷۵۰	۱۰۲/۴	۲۴	۴۱۷	۷/۵
۳۱۸	۱۹۰۲	۱۰۵/۲	۴۲	۶۸۲	۱۲
۲۰	۴۲۸	۱۱/۱	۴۰	۲۰۹	۴/۱
۵۴	۷۵۹	۲۱/۱	۹۶	۵۱۲	۱۵/۸
۶۲	۴۳۱	۲۸/۸	۴۰	۶۰۶	۱۸/۳
۲۸	۲۸۳	۱۰	۵۲	۴۰۰	۸/۹
۳۵	۲۰۵	۸	۹۴	۱۲۳۵	۳۸/۱
۳۰	۲۸۹	۴/۹	۱۱۰	۱۵۷۲	۶۱/۲
۴۸	۶۸۰	۱۲/۹	۱۵	۵۰۰	۳/۶
۹۳	۷۹۴	۱۹	۲۴	۶۹۴	۱۱/۸
۵۷	۵۱۲	۱۰/۸	۳	۱۹۹	۰/۵
۲۲	۲۲۴	۲/۹	۲۹	۲۶۰	۶/۱

www.SID.ir

Archive of SID

جدول شماره‌های ۱، ۲ و ۳مجموعه داده‌های LOC /FP و مقدار تلاش متناظر با آن‌را برای مجموعه داده ۱ تا ۳ و جدول شمارهٔ ۴ مجموعه داده LOC و مقدار تلاش متناظر با آن‌را برای مجموعه داده ۴ نشان می‌دهد.

برای استفاده از شبکه RBF برای تخمین تلاش سیستم، در آزمایش اول به ازای ورودی LOC، از مجموعه داده‌های ۱و۲، ۳۲ داده برای آموزش شبکه Newrbe و از ۷ داده برای آزمون شبکه استفاده نمودیم. جدول شمارهٔ ۵ نتایج آزمون شبکه را نشان می‌دهد. ستون اول مقدار تلاش واقعی و ستون دوم تلاش تخمینی شبکه را نشان می‌دهد. مقدار خطای حاصل ۲۸/۷۸ شده است. شکل شمارهٔ ۱ نیز، نمودار مقادیرتلاش واقعی و تخمینی را برای داده‌های آزمون شبکه نشان می‌دهد. نمودار مقادیر تلاش تخمینی با خط چین مشخص شده است.

جدول شمارهٔ ۲. مجموعه داده Kemerer (هیت، ۲۰۰۲)

تعداد خطوط کد به ۱۰۰۰	مقدار FP	تلاش واقعی (برحسب ۱۰۰۰ ساعت)
۲۵۳,۶	۱۲۱۷,۱	۴۳,۶۲
۴۰,۵	۵۰۷,۳	۱۲,۵۴
۴۵۰	۲۳۰۶,۸	۱۶۸,۳۱
۲۱۴,۴	۷۸۸,۵	۱۳,۲۱
۴۹۹,۹	۱۳۳۷,۶	۵۱,۱۲
۵۰	۴۲۱,۳	۱۲,۷۷
۴۳	۹۹,۹	۳,۵۳
۲۰۰	۹۳۳	۱۹,۸۱
۲۸۹	۱۵۹۲,۹	۱۷,۶۳
۳۹	۲۴۰	۱۰,۹۴
۲۵۴,۲	۱۶۱۱	۳۹,۳۲
۱۲۸,۶	۷۸۹	۳۵,۰۷
۱۶۱,۴	۶۹۰	۲۳,۸۶
۱۶۴,۸	۱۳۴۷,۵	۳۷,۵۳
۶۰,۲	۱۰۴۴,۳	۱۰,۶۲

www.SID.ir

Archive of SID

آموزش را با شبکه Newgrnn تکرار نمودیم و منجر به مقدار خطای ۵۴/۵۹ شد. نمودار مقادیر تلاش واقعی و تخمینی برای این شبکه در شکل شمارهٔ ۲ نشان داده شده است. در آزمایش‌های بعدی داده‌ها مورد نرمال‌سازی قرار دادیم. در آزمایش دوم به ازای ورودی Newrbe، LOC، با مجموعه داده‌های ۱ تا ۳ به استثنای ۷ داده آموزش داده شد و با ۷ داده مورد آزمون قرار گرفت. آزمایش را به ازای ورودی FP اما با شبکه Newgrnn تکرار نمودیم. جدول شمارهٔ ۶ نتایج آزمون برای LOC و FP نشان می‌دهد.

جدول شمارهٔ ۳. مجموعه داده Halmark (هیت، ۲۰۰۲)

تعداد خطوط کد به ۱۰۰۰	مقدار FP	تلاش واقعی (برحسب ۱۰۰۰ ساعت)	تعداد خطوط کد به ۱۰۰۰	مقدار FP	تلاش واقعی (برحسب ۱۰۰۰ ساعت)
۵	۷۳	۰,۵۹	۲۶,۹۳	۱۴۸	۲,۲۶
۱۱,۹۳	۱۲۱	۰,۷	۴۴	۱۷۴	۲,۲۶
۹,۸۳	۱۱۱	۰,۸۶	۱۷,۰۹	۳۴۱	۲,۳۷
۶,۹۳	۵۵	۰,۸۸	۱۹,۲۵	۶۸۴	۲,۴۴
۱۲	۹۴	۰,۹۲	۷۰,۳۹	۶۹۷	۳,۹۵
۹,۳۸	۵۹	۰,۹۶	۱۰۶	۵۰۷	۴,۰۲
۲۶,۸۲	۷۲	۱,۱۸	۳۵,۴۷	۱۷۰	۴,۱۸
۱۷,۷۴	۱۴۴	۱,۱۸	۴۹,۵۲	۳۱۴	۴,۶۶
۱۲,۷۱	۶۷	۱,۱۹	۶۶	۲۹۳	۵,۰۱
۳۵,۲۸	۸۷	۱,۳۲	۶۴,۱۸	۴۳۴	۶,۸۴
۲۵,۶۴	۳۲۰	۱,۵۵	۲۸	۷۳۸	۵۷و۷
۱۰,۳۰	۸۶	۱,۷۵	۵۰,۳۸	۱۲۰۶	۱۱,۴۲
۲۲,۹۸	۷۷	۱,۹۱	۷۲,۷۵	۷۹۱	۱۳,۱۴
۳۵,۵۰	۱۰۸	۲,۱۸	۱۲۶,۳۳	۱۲۸۴	۲۳,۳۰

www.SID.ir

Archive of SID

جدول شمارهٔ ۴. مجموعه داده ناسا (من‌زیر و همکاران، ۲۰۰۵)

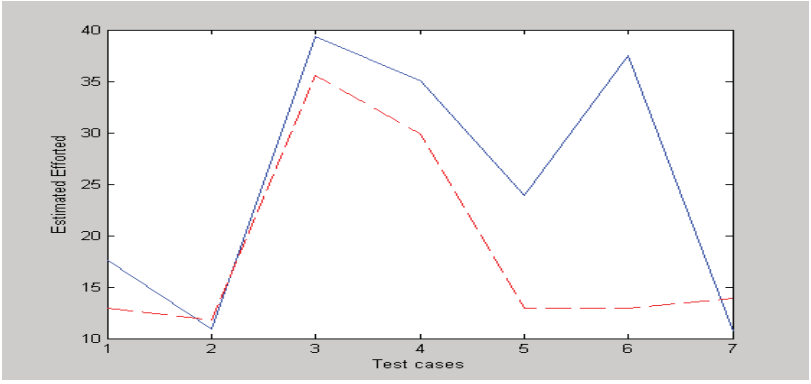
تعدادخطوط کد به ۱۰۰۰	تلاش واقعی به ماه	تعدادخطوط کد به ۱۰۰۰	تلاش واقعی به ماه
۸,۴	۲,۲	۱۲۰	۲۹,۵
۱۰,۸	۳,۵	۱۵۵	۱۹,۳
۱۸	۵,۵	۱۷۰	۳۲,۶
۲۴	۶	۱۹۲	۳۵,۵
۲۵,۲	۹,۷	۲۱۰	۳۸
۳۱,۲	۷,۷	۲۱۵	۱۰۰
۳۶	۱۱,۳	۲۳۹	۴۸,۵
۳۶	۸,۲	۲۵۲	۴۷,۵
۴۲	۶,۵	۲۷۸	۷۰
۴۲	۸	۳۰۰	۶۶,۶
۴۸	۲۰	۳۲۴	۱۵۰
۴۸	۱۰	۳۵۲,۸	۶۶,۶
۴۸	۱۵	۳۶۰	۱۰۰
۵۰	۱۰,۴	۳۶۰	۱۰۰
۶۰	۱۳	۳۷۰	۵۰
۶۰	۱۴	۴۰۰	۷۹
۶۰	۱۹,۷	۴۲۰	۱۹۰
۶۰	۳۲,۵	۴۵۰	۹۰
۶۰	۳۱,۵	۴۸۰	۱۱۵,۸
۶۲	۱۲,۸	۵۷۱,۴	۷۸
۷۰	۱۵,۴	۷۵۰	۱۰۱
۷۲	۲۰	۸۱۵	۱۶۱,۱
۷۲	۷,۵	۹۷۳	۲۸۴,۷
۸۲	۱۶,۳	۱۱۸۱	۲۲۷
۹۰	۱۵	۱۲۴۸	۱۷۷,۹
۹۸,۸	۱۱,۴	۱۳۶۸	۲۸۲,۱
۱۰۷	۲۱	۲۱۲۰	۲۱۹
۱۱۴	۱۶	۲۳۰۰	۴۲۳
۱۱۷,۶	۲۵,۹	۲۴۰۰	۳۰۲
۱۱۷,۶	۲۴,۶	۳۲۴۰	۳۷۰

www.SID.ir

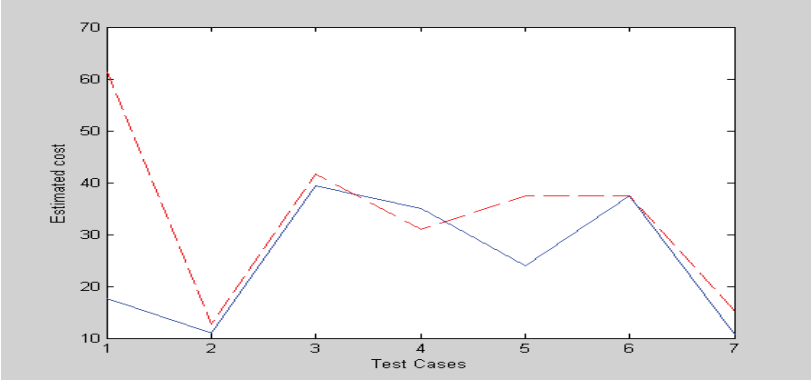
جدول شمارهٔ ۵. نتایج آزمون شبکه در آزمایش اول

تلاش واقعی	تلاش تخمینی براساس LOC
۱۷,۶۳	۱۲,۸۹
۱۰,۹۴	۱۱,۷۶
۳۹,۳۲	۳۵,۵۳
۳۵,۰۷	۲۹,۸۶
۲۳,۸۶	۱۲,۸۹
۳۷,۵۳	۱۲,۸۹
۱۰,۶۲	۱۳,۹۱
MAPE	۲۸,۷۸

شکل شمارهٔ ۱. نمودار تلاش واقعی و تخمینی آزمون اول با شبکهٔ Newrbe



شکل شمارهٔ ۲. نمودار تلاش واقعی و تخمینی آزمون اول با شبکهٔ Newgrnn



www.SID.ir

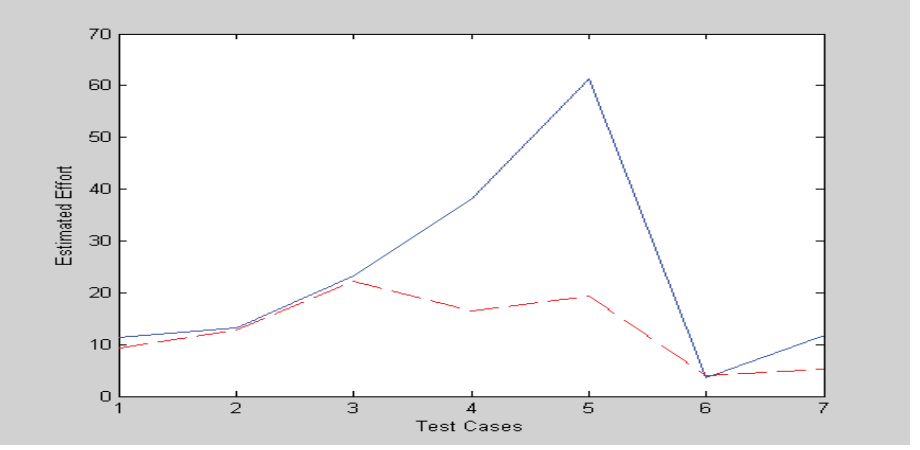
Archive of SID

جدول شمارهٔ ۶. نتایج آزمون شبکه در آزمایش دوم

تلاش واقعی	تلاش تخمینی براساس LOC	تلاش تخمینی براساس FP
۱۱,۴۲	۹,۲۱	۲۱,۴۹
۱۳,۱۴	۱۲,۸۰	۱۲,۴۶
۲۳,۳۰	۲۲,۳۰	۲۴,۱۷
۳۸,۱۰	۱۶,۴۲	۲۲,۴۴
۶۱,۲۰	۱۹,۲۷	۳۷,۵۳
۳,۶۰	۳,۹۹	۹,۳۲
۱۱,۸۰	۵,۲۶	۱۱,۲۳
خطا	۳۱,۱۱	۴۸,۶۷

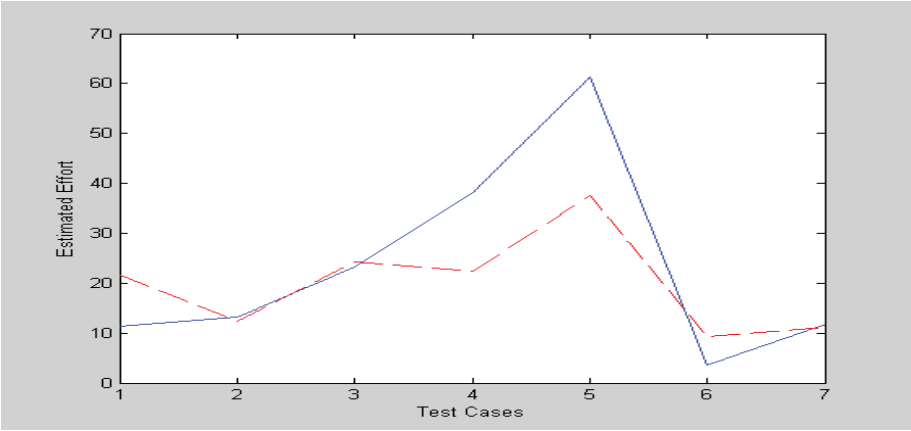
مقدار خطا به ازای LOC برابر ۱۳/۱۱ و به ازای FP برابر ۴۸/۶۷ است. شکل شمارهٔ ۳ و ۴، نمودار مقادیرتلاش واقعی و تخمینی را برای داده‌های آزمون نشان می‌دهد. نتیجه آزمایش به ازاء ورودی FP با شبکه Newrbe منجر به Mape ۵۰/۴۶ گردید. آزمایش سوم با مجموعه داده ۴ صورت گرفته است. ورودی شبکه LOC است و از شبکه Newrbe استفاده شده است. مقدار خطا برابر ۴۳/۸۲ به‌دست آمد.

شکل شمارهٔ ۳. نمودار تلاش واقعی و تخمینی آزمون دوم به ازای LOC



www.SID.ir

شکل شمارهٔ ۴. نمودار تلاش واقعی و تخمینی آزمون دوم به ازای FP



جدول۷. نتایج آزمون شبکه در آزمایش سوم

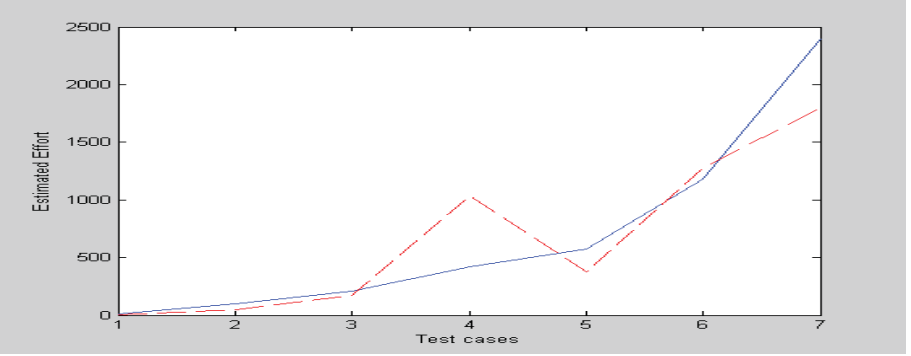
تلاش تخمینی براساس loc	تلاش واقعی
۶,۵	۸,۴
۴۴,۸	۹۸,۸
۱۷۳,۳	۲۱۰
۱۰۳۲	۴۲۰
۳۷۸,۸	۵۷۱,۴
۱۲۷۳,۵	۱۱۸۱
۱۸۰۱	۲۴۰۰
۴۳,۸۲	MAPE

شکل شمارهٔ ۵، نمودار مقادیر تلاش واقعی و تخمینی را برای داده‌های آزمون نشان می‌دهد. آزمایش سوم را با شبکه Newgrnn تکرار نمودیم. مقدار Mape برابر ۵۳/۰۱ گردید. مقدار Mape در آزمایش‌های مختلف نشان می‌دهد، تخمین تلاش با شبکهٔ عصبی RBF نسبت به روش‌های مبتنی بر الگو نتیجه بهتری را ارائه می‌دهد.

www.SID.ir

Archive of SID

شکل شمارهٔ ۵. نمودار مقادیر تلاش واقعی و تخمینی آزمایش سوم



۶. نتیجه‌گیری

در این مقاله، از شبکهٔ عصبی RBF برای تخمین تپلاش لازم برای ساخت سیستم اطلاعاتی استفاده نمودیم. همان‌گونه که در بخش ۴ اشاره شد معیار ارزیابی روش تخمین تلاش، خطا است. نتایج حاصل از بکارگیری شبکه عصبی RBF در فرایند تخمین نشان داد که خطای حاصل از شبکهٔ عصبی RBF، در مقایسه با روش‌های تخمین تلاش مبتنی بر الگو، که متداول‌ترین شیوه تخمین تلاش هستند، به مراتب کمتر است. افزون بر آن، در مراحل اولیه ساخت سیستم، دانش موجود از پروژه و در نتیجه عوامل تنظیم تلاش پروژه ناکافی بوده و امکان استفاده از روش‌های مبتنی بر الگو با تقریب مناسب وجود ندارد. بنابراین در صورت وجود مستندات مربوط به تعداد خطوط کد سیستم‌های اطلاعاتی مختلف و نفر ساعت مربوط به تلاش واقعی ساخت سیستم‌ها، می‌توان از شبکه عصبی RBF برای تخمین تلاش ساخت سیستم‌های اطلاعاتی جدید با تقریب مناسب استفاده کرد. روش پیشنهادی در این مقاله، درصوت وجود اطلاعات تلاش و هزینهٔ سیستم‌های اطلاعاتی تولید شده در معاونت فناوری اطلاعات و ارتباطات ناجا و کلیه سازمان‌های متولی تولید و توسعه سیستم‌های اطلاعاتی، قابل بهره‌برداری بوده و نقش مؤثری در تخمین تلاش و هزینهٔ لازم برای ساخت سیستم‌های اطلاعاتی خواهد داشت. به‌طور کلی و

www.SID.ir

Archive of SID

اصولی، از آنجاکه روش‌های تخمین مبتنی بر شبکهٔ عصبی براساس شیوه یادگیری عمل می‌کنند، در نتیجه درصورت وجود داده‌های مربوط به پروژه‌های تکمیل شده سیستم‌های اطلاعاتی هرسازمان، امکان استفاده از روش پیشنهادی این مقاله برای تخمین تلاش سیستم‌هایی دردست ساخت وجود دارد.

۶-۱. کارهای مرتبط

در دو دههٔ گذشته از شبکه عصبی برای پیش‌بینی در کاربردهای مختلفی استفاده شده است و نتایج حاصله از روش‌های معمول بهتر بوده است. هاکارینن و همکارانش (۱۹۹۳)، اندازه نرم‌افزار را با استفاده از شبکهٔ عصبی تخمین زدند. به این ترتیب که ورودی شبکهٔ عصبی، توصیف مشخصات ساخت یافته سیستم‌ها، و خروجی آن مقدار FP سیستم بود. بررسی نتایج نشان داد که شبکهٔ عصبی می‌تواند به‌صورت موفق برای تخمین تلاش اندازه نرم‌افزار استفاده شود.

میر و همکارانش (۲۰۰۰)، سه روش مبتنی بر یادگیری، یعنی شبکهٔ عصبی، استدلال مبتنی برمورد[1] ویادگیری استقرایی قوانین با استفاده از داده ۷۷ پروژه مورد بررسی قرار گرفته است. در این بررسی از شبکه MLP با الگوریتم یادگیری پس انتشار خطا استفاده شده است. ورودی شبکهٔ ۱۰، ویژگی پروژه بوده است. نتایج این بررسی نشان داد که دقت شبکهٔ عصبی از سایر روش‌ها بهتر بوده است.

1. Case-based Reasoning

www.SID.ir

Archive of SID

۷. منابع

1. G Kim,et al, "Comparison Of Construction Cost Estimating Models Based On Regression Analysis, Neural Networks, And Case-Based Reasoning," Elsevier 2004.
2. B. Barry, C. Abts, S. Chulani, "Software Development Cost Estimation Approaches—A Survey,» Annals of Software Engineering, 2000.
3. B.Boehm,»Software Engineering Economics,» Englewood Clifs, NJ, Prentice-Hall, 1981.
4. F.Heemstra,» Software cost estimation,» Information and Software Technology, pp. 627-639, 1992.
5. A.Gray, S. MacDonell, «A Comparison Of Techniques For Developing Predictive Models Of Software Metrics,»Information and Software Technology 39, 1997.
6. A. Idri,et al, “Estimating Software Project Effort By Analogy Based On Linguistic Values,” Proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS.02)0-7695-1339-5/ 2002 IEEE.
7. B. Barbro, “Neural Networks And Genetic Algorithms For Bankruptcy Predictions,” Expert Systems with Application 11 (4), 407–413, 1996.
8. M.J.A.Berry,G.Linoff,”Data Mining Techniques For Marketing Sales And Customer Support,” Wiley,New York, 1997.
9. J.P. Bigus, “Data Mining with Neural Networks,” McGraw-Hill, New York, 1996.
10. A. Heiat, “Comparison Of Artificial Neural Network And Regression Models For Estimating Software Development Effort,” Information and Software Technology 44 , Elsevier, 2002.
11. A.J. Albrecht, J. Gaffney, “Software Function, Source Lines Of Code, And Development Effort Prediction: A Software Science Validation,” IEEE Transactions on Software Engineering SE-9 (6), 639–648, 1983.
12. B.Boehm, et al, “Cost Model For Future Software Life Cycle Processes:

www.SID.ir

Cocomo2," Annals of software Engineering,1995.

13. H. Demuth, M. Beale, "Neural network toolbox for use with MATLAB, user's guide version 4, neural network toolbox user's guide," copyright 1992 - 2002, MathWorks, Inc.
14. C.F. Kemerer, "An Empirical Validation Of Software Cost Estimation Models,» Communications of the ACM 30 416–429, 1987.
15. D.V. Ferens, R.B. Gumer, «An Evaluation For Three Function Point Models Of Estimation Of Software Effort,» IEEE National Aerospace and Electronics Conference, 1992.
16. S.S.Vicinanza, T.Mukhopadhyay, J.J. Prietula, "Software-Effort Estimation: An Exploratory Study Of Expert Performance,» Information Systems Research, December , 243–262, 1991
17. T.Menzies,et al,» Validation Methods For Calibrating Software Effort Models, « 27th International conference on Software Engineering, Missouri,USA,15-21,2005.
18. J. Hakkarainen, et al "Neural Networks In Specification Level Software Size Estimation," Proceeding of the Twenty-Sixth Hawaii International Conference on Volume iv, 5-8 Jan. 1993 Page(s):626 - 634 vol.4.
19. C.Mair,et al, "An Investigation Of Machine Learning Based Prediction Systems," Journal of systems and software,volume 53,Issue 1, 2000.

www.SID.ir