یادگیری ارجحیت با استفاده از شبکه‌های عصبی مصنوعی

داریوش حسن‌پور آده[۱]

[۱] دانشگاه صنعتی اصفهان، دانشکده مهندسی برق و کامپیوتر، شماره دانشجویی: ۹۳۰۸۱۶۴

چکیده

یادگیری‌ارجحیت یکی از زیررشته‌های یادگیری‌ماشین می‌باشد که هدف اصلی‌اش یادگیری ارجحیت‌های قابل پیش‌بینی از روی اطلاعات ارجحیت می‌باشد. از نقطه‌نظر یادگیری باناظر یادگیری‌ارجحیت روی یک دسته از عناصر که نسبت به یک دسته از برچسب‌ها یا عناصر ارجحیت بیشتری دارند آموزش داده می‌شوند که بتواند در نهایت مدل ارجحیت عناصر دیده نشده را پیش‌بینی کند. در این گزارش با تکیه بر مطالب مقاله‌ی اصلی به معرفی یک روش یادگیری جهت یادگیری رتبه‌بندی[۱] می‌پردازیم که توسط شبکه‌ای به نام CmpNN به رتبه‌بندی میان اشیا می‌پردازد. شبکه‌ی معرفی شده دارای معماری خاصی بوده و الگوریتمی که معرفی شده به نحو خاصی آموزش داده می‌شود تا بهترین توصیف آموزشی متناسب با وزن‌های شبکه و داده‌های آموزشی را مشاهده کند. همچنین در کنار شرح مقاله‌ی اصلی به مروری بر دیگر کارهای انجام شده در زمینه‌ی یادگیری‌ارجحیت با شبکه‌های عصبی مصنوعی خواهیم پرداخت.

کلمات کلیدی

یادگیری ارجحیت، شبکه‌های عصبی مصنوعی، شبکه‌ی ارجحیت، یادگیری انتخابی، طبقه‌بندی

۱ مقدمه

یادگیری ارجحیت[۲] یکی از شاخه‌های یادگیری‌ماشین[۳] می‌باشد که در سال های اخیر توجه زیادی را به سمت خود جلب کرده است. وظیفه‌ی اصلی یادگیری ارجحیت، یادگیری رتبه‌بندی کردن عناصر می‌باشد. که با توجه به نوع اطلاعات آموزشی نوع ارجحیت و همچنین نوع رتبه‌بندی می‌تواند تغییر کند. در حالت کلی یادگیری‌ارجحیت به استنتاج کردن ارتباطات بین اعضای یک دسته و نگاشت این ارتباطات به مدلی که بتواند ارجحیت نسبی موجود بین اعضای این دسته بخوبی پیش‌بینی کند.

اگر بخواهیم تعریف دقیقی برای یادگیری‌ارجحیت ارائه دهیم می‌توانیم بگوییم که یادگیری‌ارجحیت به وظیفه‌ی یادگیری پیش‌بینی کردن رابطه‌ی ترتیب[۴] در مجموعه‌ای از اشیا گفت[۴]. یادگیری‌ارجحیت از داده‌های اطلاعات ارجحیت[۵] برای اهداف یادگیری خود استفاده می‌کند که این اطلاعات ارجحیت نقش مهمی در تصمیم‌گیری‌های خودکار در زمینه‌های متعددی مانند تئوری تصمیم‌گیری های کیفی[۶]، استدلال غیریکنواخت[۷]، ارضای محدودیت[۸] و طرح‌ریزی[۹] و ... کاربردهای فراوانی دارد. در مرحله‌ی آموزش، الگوریتم‌های یادگیری‌ارجحیت به داده‌های ترتیب رتبه‌بندی[۱۰] (نیمه)شناخته شده عناصر دسترسی دارند. بسته به مدل و نوع مساله الگوریتم‌های یادگیری‌ارجحیت می‌توانند به ۳ دسته‌ی رتبه‌بندی اشیا[۱۱]، رتبه‌بندی برچسب‌ها[۱۲] و رتبه‌بندی نمونه‌ها[۱۳] تقسیم‌بندی کرد.

در این نوشتار ابتدا مقدمه‌ای بر مفاهیم بنیادی مطرح در یادگیری‌ارجحیت آورده شده است و بعد از مروری خلاصه بر تعدادی از کارهای انجام شده در این زمینه، با تمرکز بر مقاله اصلی یادگیری‌ارجحیت با استفاده از ANN را ارائه خواهیم داد.

۲ یادگیری‌ارجحیت

برای اینکه بتوانیم یادگیری‌ارجحیت با استفاده از شبکه‌های عصبی مصنوعی را ارائه دهیم نیاز هست که در این قسمت به معرفی خلاصه‌ای از مفاهیم اولیه و بنیادی زمینه‌ی یادگیری‌ارجحیت بپردازیم.

۲-۱ نمادگذاری

طبق هر شاخه‌ی علمی دیگر به یک سری نماد برای پایه‌گذاری یادگیری ارجحیت نیاز داریم که در این قسمت به معرفی آن‌ها می‌پردازیم.

تعریف ۱ (ارجحیت ضعیف). یک ارجحیت ضعیف با نماد $\succeq$ بروی مجموعه‌ای مانند A یک رابطه‌ی بازتابی و تعددی می‌باشد.

تعریف ۲ (ارجحیت اکید). $a \succ b \leftrightarrow (a \succeq b) \wedge (b \not\succeq a)$

در راستای معناشناسی یادگیری‌ارجحیت تعاریف ۱ و ۲ می‌توان آن‌ها را به صورت جدول ۱ تعبیر کرد.

نماد	تعبیر
$a \succeq b$	"جایگزین a حداقل به اندازه‌ی جایگزین b ترجیح داده می‌شود."
$a \succ b$	"جایگزین a بیشتر از جایگزین b ترجیح داده می‌شود."

جدول۱: تعبیر نمادهای روابط ارجحیت

تعریف ۳ (ترتیب‌اکید کلی[۱۴] - رتبه‌بندی[۱۵]). اگر $\mathcal{A}$ یک مجموعه‌ای از اشیا / جایگزین‌ها[۱۶] $\{a_1, \ldots, a_m\}$ باشد، یک رتبه‌بندی از $\mathcal{A}$ یک جایگشتی همانند τ از مجموعه‌ی $\{1, \ldots, m\}$ می‌باشد بگونه‌ای که $a_i \succ a_j \leftrightarrow \tau(i) < \tau(j)$.

در تعریف ۳ مقدار m تعداد اشیا/جایگزین‌ها و a_iها همان اشیا/جایگزین‌ها می‌باشند، که از این به بعد به مجموعه‌ی اشیا یا جایگزین‌ها، فقط جایگزین گفته خواهد شد. تعریف ۳ در واقع فرمت[۱۷] خروجی کلی الگوریتم‌های یادگیری‌ارجحیت را ارائه می‌دهد، به گونه‌ای که خروجی ارجحیت ورودی‌ها یک جایگشتی اندیسی[۱۸] ترتیب‌اکید ارجحیت‌های ورودی‌ها می‌باشد بگونه‌ای که **اندیس** آن جایگزینی که **دارای ارجحیت بیشتری** نسبت به دیگری است، در مجموعه‌ی τ **مقدم‌تر** از دیگری می‌آید.

طبق آنچه که در تعریف ۳ آمده، واضح است که الگوریتم‌های یادگیری ارجحیت درواقع یک جستجوکننده در فضای جایگشتی مجموعه τ می‌باشد؛ فضای جایگشت‌های τ را با $\mathcal{S}_m$ نمایش می‌دهند.

۲-۲ انواع الگوریتم‌های رتبه‌بندی

الگوریتم‌های رتبه‌بندی از نظر نوع وظیفه‌ای که به عهده دارند به ۳ تیپ تقسیم بندی می‌شوند که عبارتند از رتبه‌بندی برچسب‌ها، رتبه‌بندی اشیا و رتبه‌بندی نمونه ها؛ که در این قسمت شرح مختصری از این ۳ تیپ الگوریتم ارائه خواهیم داد[۴].

۲-۲-۱ رتبه‌بندی برچسب‌ها

در این رتبه‌بندی هدف این است که به ازای یک نمونه برچسب‌ها را براساس ارجحیت مرتب کند. شماتیک ورودی‌ها و خروجی‌ها این نوع الگوریتم در شکل ۱ آمده است.

Given:

- A set of training instances $\{x_k \mid k=1,\ldots,n\} \subseteq \mathcal{X}$ (each instance typically represented by a feature vector)
- A set of labels $\mathcal{L} = \{\lambda_i \mid i = 1,\ldots,m\}$
- For each training instance x_k: a set of associated *pairwise preferences* of the form $\lambda_i \succ_{x_k} \lambda_j$

Find:

- A ranking function in the form of an $\mathcal{X} \longrightarrow \mathcal{S}_m$ mapping that assigns a ranking (permutation) $\succ_x$ of $\mathcal{L}$ to every $x \in \mathcal{X}$

شکل۱: شماتیک کلی الگوریتم‌های رتبه‌بندی برچسب‌ها

همانطور که در شکل ۱ آمده است الگوریتم‌های رتبه‌بندی برچسب‌ها تعدادی نمونه‌ی آموزشی می‌گیرد که این نمونه‌های آموزشی معمولا به صورت بردار ویژگی ها نمایش داده می‌شود. سپس یک سری برچسب و همچنین به ازای هرکدام از نمونه‌ها یک مجموعه‌ی دوبه‌دو مرتب که ارجحیت برچسب‌ها برای هرکدام از نمونه ها مشخص شده است را به عنوان ورودی به الگوریتم داده می‌شود.الگوریتم‌های متعلق به رتبه‌بندی برچسب‌ها باید به عنوان خروجی یک تابع رتبه‌بند است که بتواند به ازای یک شی جایگشت‌های برچسب‌های متعلق به آن شی را با توجه به ارجحیتی که دارند برگرداند.

۲-۲-۲ رتبه‌بندی اشیا

در این رتبه‌بندی هدف این است که یک دسته از اشیا را برحسب ارجحیتی که نسبت به هم دارند، را مرتب‌سازی کنیم. که شماتیک ورودی‌ها و خروجی‌ها این نوع الگوریتم در شکل ۲ آمده است.

Given:

- A (potentially infinite) set $\mathcal{X}$ of objects (each object typically represented by a feature vector)
- A finite set of pairwise preferences $x_i \succ x_j$, $(x_i, x_j) \in \mathcal{X} \times \mathcal{X}$

Find:

- A ranking function $r(\cdot)$ that, given a set of objects $\mathcal{O} \subseteq \mathcal{X}$ as input, returns a permutation (ranking) of these objects

شکل۲: شماتیک کلی الگوریتم‌های رتبه‌بندی اشیا

در شماتیکی که در شکل ۲ آمده است می‌بینیم که همانند رتبه‌بندی برچسب‌ها یک دسته از اشیا را به عنوان ورودی می‌گیرد. سپس یک دسته‌ی (تکه‌ای)مرتب از ارجحیت‌های نسبی این اشیا را نیز به عنوان ورودی به الگوریتم می‌دهیم و نهایت الگوریتم باید یک تابع رتبه‌بند را بیابد که بتواند ارجحیت‌های هر زیرمجموعه از مجموعه‌ی جهانی اشیا را در قالب یک جایگشتی از اشیا ورودی به عنوان خروجی برگرداند.

توجه شود که مجموعه‌ی (تکه‌ای)مرتب از ارجحیت اشیا که به عنوان داده آموزشی به الگوریتم داده می‌شود نیازی ندارد که تمامی ارجحیت‌های موجود میاد اشیا را شامل باشد، به چند دلیل این شرط می‌تواند به قابل پیاده‌سازی بودن الگوریتم کمک زیادی کند. اولین دلیل که بدیهی‌ترین دلیل می‌باشد، این است که ممکن است در کابردهای دنیای واقعی این الگوریتم تمامی ارجحیت های موجود بین اشیا نشاخته نشده باشد. دومین دلیل این است که در صورتی که تمامی ارجحیت های بین اشیا شناخته شده باشد اندازه‌ی مجموعه‌ی مرتب از ارجحیت خیلی بیشتر از تعداد اشیا می‌باشد؛ که از حل ۱ می‌بینیم این افزونگی داده‌ای تقریبا $\frac{n}{2}$ برابر تعداد اشیا می‌باشد که در کابردهای واقعی این الگوریتم‌ها معمولا تعداد اشیا زیاد است که در نتیجه تعداد ارتباطات مرتب بین آنها بسیار بیشتر از تعداد اشیا می‌باشد که مشکلات خواص خودش را در پی دارد که خارج از حیطه‌ی این نوشتار است.

$$\frac{\binom{n}{2}}{n} = \frac{n-1}{2} \qquad (۱)$$

در نتیجه این شرط که الگوریتم‌های رتبه‌بندی اشیا تنها با در اختیار داشتن مجموعه ی تکه‌ای مرتب از ارجحیت های میان اشیا باید بتواند مدلی برای رتبه‌بندی اشیا بدست بیاورد؛ باعث می‌شود که این الگوریتم‌های در کاربردهای دنیای واقعی کاربرد

پیدا کند.

۲-۲-۳ رتبه‌بندی نمونه

رتبه‌بندی نمونه همانند رتبه‌بندی اشیا می‌باشد یعنی علاوه بر شماتیک معرفی شده در رتبه‌بندی اشیا ۲ عدد ورودی دیگر را نیز دارد. به این صورت که یک مجموعه اکیدا مرتب(تعریف ۲ را ببینید) از برچسب‌ها را نیز اختیار می‌کند که در این مجموعه مرتب از برچسب‌ها آن برچسبی که دارای ارجحیت بیشتری است در ابتدای مجموعه ظاهر شود. و همچنین یک ورودی دیگر نیز به الگوریتم داده می شود که مجموعه ارتباطات یک‌به‌یک از هرکدام از اشیا را به برچسب متعلق به آن شی می‌باشد. همچنین در این نوع از الگوریتم‌ها نیازی به مجموعه (تکه‌ای)مرتب از ارجحیت‌های نسبی اشیا به الگوریتم داده شود.

۲-۳ موارد خاص یادگیری‌ارجحیت

همان‌طور که در قسمت ۲-۲ آمده است الگوریتم‌های یادگیری‌ارجحیت از نظر اهداف و کاربرد با یک‌دیگر متفاوت هستند، ما همه روزه در میان الگوریتم‌های طبقه‌بندی روزمره‌ای که با آنها سروکار دارم در واقع داریم از الگوریتم‌های ارجحیت استفاده می‌کنیم که در اینجا به معرفی دو نوع از الگوریتم‌های طبقه‌بندی می پردازیم که حالت خاصی از الگوریتم‌های یادگیری‌ارجحیت می‌باشند.

۲-۳-۱ طبقه‌بندی

الگوریتم‌های طبقه‌بند[۱۹] که همه روزه از آن‌ها استفاده می‌کنیم در واقع حالت خاصی از الگوریتم‌های رتبه‌بندی برچسب‌ها می‌باشد. از آنجایی که در طبقه‌بندی به ازای یک ورودی هدف پیش‌بینی برچسب مرتبط با آن ورودی می‌باشد، می‌توان داده ی ورودی را به یکی از الگوریتم‌های یادگیری‌ارجحیت (رتبه‌بندی برچسب‌ها) داد سپس از میان مجموعه جایگشت برچسب‌ها که به عنوان خروجی رتبه‌بند برگشت داده خواهد شد، آن برچسبی را که دارای بیشتری ارجحیت است را به عنوان برچسب منتخب برای آن نمونه ورودی انتصاب کرد. به عبارت دیگر می‌توان با استفاده از تعریف ۲ الگوریتم‌های طبقه‌بند را با استفاده از رتبه‌بند برچسب‌های یادگیری ارجحیت مدل کرد.

$$\mathcal{C}_{x_k} = \{\lambda_i \mid \lambda_i \succ_{x_k} \lambda_j,\ 1 \le j \ne i \le m\} \qquad (۲)$$

۲-۳-۲ طبقه‌بندی چند برچسب

در طبقه‌بندی چند برچسب[۲۰] هر نمونه به یک زیرمجموعه‌ای از برچسب‌ها نسبت داده‌می‌شود. این نوع از الگوریتم‌ها را نیز حالت خاصی از الگوریتم‌های یادگیری‌ارجحیت (رتبه‌بندی برچسب‌ها) می‌باشد به‌گونه‌ای که الگوریتم رتبه‌بندی برچسب‌ها یک عدد به عنوان تعداد برچسب‌های نسبت داده شده به اشیا داده می شود و الگوریتم بعد از مرتب‌سازی برچسب‌ها بر اساس ارجحیت‌هایی که دارند یک تعداد از برچسب‌ها که دارای ارجحیت بیشتری می‌باشند را به عنوان خروجی بر می گرداند. یا یک روش دیگر برای مدل کردن طبقه‌بند چند برچسب با استفاده از رتبه بند برچسب‌ها این است که در کنار داده‌های الگوریتم یک حد آستانه مابین $[0,\ 1]$ به الگوریتم داده شود و بعد از رتبه‌بندی برچسب‌ها و نرمال‌سازی میزان ارجحیت های برچسب‌ها آن برچسب‌هایی که میزان ارجحیت آن‌ها بیشتر از حد آستانه به عنوان خروجی برگشت داده شود. در حالت کلی می‌توان طبقه‌بند چند برچسب را به صورت تعریف ۳ با استفاده از رتبه‌بند ارجحیت مدل کرد.

$$\mathcal{C}_{x_k} = \{\mathcal{L}_k \mid \lambda_i \succ_{x_k} \lambda_j,\ \lambda_i \in \mathcal{L}_k, \lambda_j \in \mathcal{L}\backslash\mathcal{L}_k\} \qquad (۳)$$

۲-۴ انواع روش‌های یادگیری‌ارجحیت

برای یادگیری‌ارجحیت دو روش معمول موجود است که یکی بر اساس یادگیری یک تابع سودمندی[۲۱] و دیگری یادگیری ارتباطات ارجحیت[۲۲] موجود بین داده‌ها را یاد می‌گیرید که برای درک بهتر روش ارائه شده در مقاله اصلی در این بخش به توضیح مختصری نسبت به هریک می‌پردازیم.

۲-۴-۱ یادگیری تابع سودمندی

یک راه طبیعی برای نشان دادن ارجحیت‌های موجود بین داده‌ها این‌است که جایگزین‌های موجود را با استفاده از یک تابع مطلوبیت ارزیابی کنیم. برای رتبه‌بندی اشیا یک تابع نگاشت $f : \mathcal{X} \to \mathcal{U}$ که مقدار سودمندی $f(x)$ به هریک از اشیا x تخصیص می‌دهد را یاد می‌گیریم و درنهایت اشیا را با معیار سودمندی‌ای که از این تابع سودمندی یادگرفته شده محاسبه می‌شود مرتب کرده و به عنوان خروجی برمی‌گردانیم. و در رتبه‌بندی برچسب‌ها به ازای هریک از برچسب‌ها $\lambda_i,\ i = 1,\ \ldots,\ m$ یک تابع سودمندی $f_i : \mathcal{X} \to \mathcal{U}$ یادگرفته می‌شود و در نهایت برچسب‌ها را برحسب مقدار سودمندی‌ای که آن برچسب‌ها دارند مرتب می‌کنیم، $\lambda_i \succ_x \lambda_j \Rightarrow f_i(x) \ge f_j(x)$.

۲-۵ یادگیری ارتباطات ارجحیت

یک روش معمول دیگر در یادگیری‌ارجحیت این است که بیاییم و ارتباطات ارجحیت دودویی موجود بین دو اشیا(برچسب) ها را یاد بگیریم. برای رتبه‌بندی اشیا یک تابع ارجحیت دودویی مانند $\mathcal{Q}(x,\ x\prime)$ یاد می‌گیرد که نشان می‌دهد آیا شی x نسبت به $x\prime$ ارجحیت دارد یا خیر. در نهایت ترتیب نهایی شامل یک جایگشتی می‌باشد که حداکثر سازگاری را با این ترتیب‌های دودویی داشته باشند. به عبارت دیگر جایگشتی را انتخاب می‌کنیم که مقدار ۴ را حداکثر کند.

$$V_{\mathcal{X}} = \sum_{i=1}^{n} \sum_{j \ne i=1}^{n} \mathcal{Q}(x_i, x_j) \qquad (۴)$$

برای رتبه‌بندی برچسب‌ها هم یک روشی به‌نام طبقه‌بندی دوبه‌دو ارائه شده‌است. ایده‌ی این طبقه‌بند دوبه‌دو این است که به ازای هریک از برچسب‌ها یک مدلی مانند $\mathcal{M}_{i,j},\ 1 \le i < j \le m,\ (\lambda_i,\ \lambda_j) \in \mathcal{L}$ را با استفاده از داده‌های آموزشی بدست بیاوریم(یاد بگیریم)؛ در نتیجه به تعداد $\frac{m(m-1)}{2}$ تعداد مدل نیاز داریم. داده‌های آموزشی شامل اطلاعات ارجحیت برچسب‌ها مانند $\lambda_i \succ_x \lambda_j$ می‌باشند که به‌صورت مثال‌های آموزشی $(a,\ b)$ تبدیل می‌شوند که برای یادگیری مدل $\mathcal{M}_{a,\ b},\ a = \min(i,j),\ b = \max(i,j)$ استفاده می‌شوند. در واقع مدل $\mathcal{M}_{a,b}$ سعی بر یادگیری مدلی دارد که نگاشت ۵ را انجام دهد.

$$\mathcal{M}_{a,b} \mapsto \begin{cases} 1 & \lambda_a \succ_x \lambda_b \\ 0 & \lambda_b \succ_x \lambda_a \end{cases} \qquad (۵)$$

که این نگاشت می‌تواند توسط یکی از طبقه‌بند کننده‌های دودویی انجام گیرد.

۳ یادگیری‌ارجحیت با استفاده از شبکه‌های عصبی

در این قسمت به شرح یادگیری‌ارجحیت با استفاده از شبکه‌های عصبی مصنوعی می‌پردازیم، ابتدا به معرفی شبکه‌ای جهت مقایسه کردن دو نمونه و سپس الگوریتمی جهت آموزش این شبکه با استفاده از داده‌های آموزشی که بتواند بین نمونه‌های ارجحیت قائل شود می‌پردازیم و در نهایت به ارائه نتایج آزمایش‌ها پرداخته و با یک نتیجه‌گیری به این نوشتار خاتمه می‌بخشیم.

۳-۱ شبکه‌ی CmpNN

همان‌طور که گفته شد جهت تعیین ارجحیت بین اشیا باید بتوان به روشی دو نمونه را نسبت به هم مقایسه کرد که یکی از روش‌های مقایسه یادگیری توابع ارزیاب می‌باشد، تابع ارزیابی که معرفی شده است یک شبکه‌ی عصبی انتشار-به-جلو[۲۳] دولایه می‌باشد که از معماری خاصی پیروی می‌کند. در این روش دو نمونه به صورت یک زوج مرتب آموزش داده به شبکه داده می‌شود و از خروجی میزان ارجحیت موجود بین این دو نمونه برگردانده می‌شود. در اینجا فرض بر این شده است که هر نمونه ورودی شبکه به تعداد d بعد دارند؛ بنابرین از آنجایی که هر دو نمونه جهت مقایسه به شبکه داده می‌شود، لذا شبکه باید دارای $2d$ عدد ورودی باشد، همچنین یک لایه‌ی مخفی و ۲ خروجی برای شبکه در نظر گرفته شده است.

اگر فرض کنیم هر نمونه‌ی x موجود در دیتاست دارای d بعد داشته باشد:

$$\forall x \in \mathcal{D} \Rightarrow x : [x_1, \dots, x_d]^T$$

ورودی‌های شبکه‌ی معرفی شده به ازای دو نمونه از دیتاست جهت مقایسه به صورت زوج مرتب $[x, y]$ نمایش داده می‌شود بطوری که:

$$[x, y]^T : [x_1, \dots, x_d, y_1, \dots, y_d]^T$$

و دو خروجی شبکه که به صورت $N_{\succ}([x, y])$ و $N_{\prec}([x, y])$ نمایش داده می شود، که تفسیر خروجی‌های شبکه به شرح زیر می‌باشد:

- $N_{\succ}([x, y])$ میزان نرخ ارجحیت x نسبت به y را نشان می‌دهد.
- $N_{\prec}([x, y])$ میزان نرخ ارجحیت y نسبت به x را نشان می‌دهد.

این شبکه‌ی عصبی می‌تواند توسط الگوریتم‌های معمول آموزش شبکه همانند الگوریتم انتشار-به-عقب آموزش داده شود. برای داده‌های آموزشی نیز به ازای هر جفت نمونه‌ای که ارجحیت آن‌ها را نسبت به هم می‌دانیم مقادیر هدف خروجی های شبکه را بصورت زیر مقدار دهی می‌کنیم:

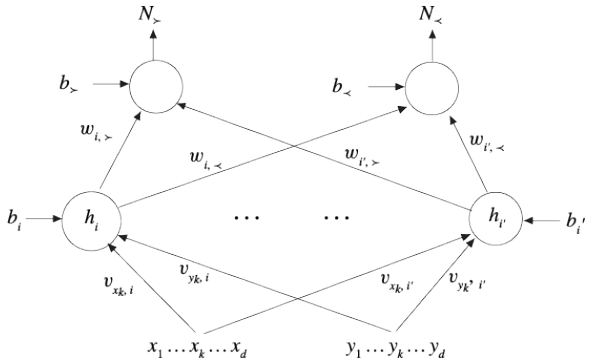
$$t = [x, y] = \begin{cases} \begin{bmatrix} 1 & 0 \end{bmatrix}^T & x \succ y \\ \begin{bmatrix} 0 & 1 \end{bmatrix}^T & y \succ x \end{cases} \tag{۶}$$

و تابع خطا نیز توسط خطای مربعات خروجی‌ها محاسبه می‌شود:

$$E([x,\ y],\ t) = (t_1 - N_{\succ}([x, y]))^2 + (t_2 - N_{\prec}([x, y]))^2 \tag{۷}$$

که بعد از آموزش مدل، می‌توانیم از شبکه برای پیش‌بینی میزان ارجحیت بین دو نمونه استفاده کرد به گونه‌ای که:

$$\forall x, y \in \mathcal{D} : \begin{cases} x \succ y & N_{\succ}([x, y]) \geq N_{\prec}([x, y]) \\ y \succ x & N_{\succ}([x, y]) \leq N_{\prec}([x, y]) \end{cases} \tag{۸}$$



شکل۳: شبکه‌ی معرفی شده‌ی CmpNN جهت مقایسه‌ی ارجحیت بین نمونه‌ها

یعنی اینکه اگر $N_{\succ}([x, y]) \geq N_{\prec}([x, y])$ آن‌گاه x حداقل به اندازه‌ی y ارجحیت دارد و برعکس.

در یادگیری‌ارجحیت توابع ارزیابی که میزان ارجحیت بین نمونه‌ها را تعیین می کنند باید شامل خواص زیر باشند:

۱. بازتابی[۲۴]: $\forall x \in \mathcal{D} \Rightarrow x \succeq x \ \& \ x \preceq x.$
۲. هم‌ارزی بین $\succ$ و $\prec$: اگر $x \succ y$ آن‌گاه $y \prec x$.
۳. پادتقارن[۲۵]: اگر $x \succ y$ و $y \succ x$ آن‌گاه $x = y$.
۴. متعدی: اگر $x \succ y$ و $y \succ z$، آن‌گاه $x \succ z$.

روش‌ارائه شده خواص ۱ و ۲ را ارضا می‌کند ولی در شرایطی که $N_{\succ}([x, y]) = N_{\prec}([x, y])$ و $x \neq y$ خاصیت سوم را نمی‌تواند ارضا کند و همچنین ادعا کرده است که خاصیت چهارم را هیچ یک از روش‌های یادگیری‌ارجحیت نمی‌تواند گارانتی کند که همیشه می‌تواند ارضا کند(روش ارائه شده نیز نمی‌تواند این شرط را ارضا کند). با وجود اینکه خاصیت چهارم یک خاصیت مهمی می‌باشد که برای اینکه الگوریتم‌های یادگیری‌ارجحیت بتواند بخوبی بین نمونه‌ها ارجحیت قائل شوند باید دارا باشند ولی در از آن‌جایی که معمولا توابع ارزیابی همچون CmpNN توسط یک الگوریتم مرتب‌سازی جهت مقایسه و سپس مرتب کردن نمونه‌ها مورد استفاده قرار می‌گیرد، در همه‌ی الگوریتم‌های مرتب‌سازی معمول خاصیت چهارم به‌صورت پیش‌فرض در نظر گرفته شده است. در نتیجه عدم ارضای این خاصیت چهارم در درست عمل‌کردن این شبکه‌ی ارزیاب برای رتبه‌دهی و ارجحیت‌بندی نمونه ها تاثیری نخواهد داشت(به شرط آن‌که روش مرتب‌سازی که این شبکه را به عنوان مقایسه‌کننده‌ی بین نمونه‌ای استفاده می‌کند خاصیت چهارم را به‌صورت پیش‌فرض درنظر داشته باشد).

شبکه‌ی CmpNN معرفی شده در شکل ۳ آمده است. شبکه‌های عصبی در حالت عمومی خواص ۱ و ۲ را نمی‌توانند ارضا کنند لذا در توضیح این روش آمده است که برای این‌که خواص ۱ و ۲ را بتوانند ارضا شوند باید از تکنیک اشتراک گذاری وزن‌ها استفاده شود. با توجه به نماد گذاری مقاله‌ی اصلی $v_{x_k,i}(v_{y_k,i})$ به معنی این است که یال متصل از ورودی x بعد kام به نورون iام لایه‌ی مخفی و $w_{i,\succ}(w_{i,\prec})$ یال متصل بین نورون iام لایه‌ی مخفی به خروجی $\succ$ شبکه می باشند(به شکل ۳ توجه کنید).

برای اشتراک گذاری وزن‌ها هر دو وزن در هر لایه وزن‌هایشان به اشتراک گذاشته می‌شوند. به‌این صورت که برای وزن‌های بین ورودی‌ها و لایه‌ی مخفی

Algorithm 1 SortNet learning algorithm

1: $T \leftarrow Set\ of\ training\ objects$
2: $V \leftarrow Set\ of\ validation\ objects$
3: $P_T^0 \leftarrow \emptyset$;
4: $P_V^0 \leftarrow \emptyset$;
5: $C^0 \leftarrow RandomWeightNetwork()$;
6: **for** $i = 0$ to max_iter **do**
7: **if** $i \geq 1$ **then**
8: $C^i \leftarrow TrainAndValidate(P_T^i, P_V^i)$;
9: **end if**
10: $[E_T^i\ ,\ R_T^i] \leftarrow Sort(C^i, T)$;
11: $[E_V^i\ ,\ R_V^i] \leftarrow Sort(C^i, V)$;
12: $score \leftarrow RankQuality(R_V^i)$;
13: **if** $score > best_score$ **then**
14: $best_score \leftarrow score$;
15: $C^* \leftarrow C^i$;
16: **end if**
17: $P_T^{i+1} \leftarrow P_T^i \cup E_T^i$;
18: $P_V^{i+1} \leftarrow P_V^i \cup E_V^i$;
19: **if** $P_T^{i+1} = P_T^i$ and $P_V^{i+1} = P_V^i$ **then**
20: return C^*;
21: **end if**
22: **end for**
23: return C^*;

شکل۴: الگوریتم SortNet

$\upsilon_{x_k,i'} = \upsilon_{y_k,i}$ و $\upsilon_{y_k,i'} = \upsilon_{x_k,i}$ یعنی وزن‌هایی بین دو نورون از لایه‌ی مخفی و دو عدد از ورودی‌های(دو بعد از دو نمونه‌ی x و y) با هم به اشتراک گذاشته می‌شوند؛ همچنین بایاس‌های لایه‌ی مخفی نیز اشتراک وزن دارند و در مورد وزن های بین لایه‌ی مخفی و خروجی‌ها به در دو وزن از دو نورون متفاوت از لایه‌ی مخفی به دو خروجی به اشتراک گذاشته می‌شود(همانند کاری که در اشتراک وزن بین ورودی‌ها و لایه‌ی مخفی صورت گرفت).

۳-۲ یادگیری رتبه‌بندی با SortNet

اکنون که با شبکه‌ای که می‌تواند ارجحیت بین داده‌ها را یادبگیرد آشنا شدیم، به معرفی الگوریتم جهت آموزش این شبکه که باعث می‌شود شبکه‌ی CmpNN بهترین وزن‌های ممکن متناسب با داده‌های را داشته باشد که در نهایت باعث می شود که ارجحیت‌بندی بهتری داشته باشیم. الگوریتمی که در اینجا معرفی می‌شود SortNet نام دارد، که مرتبه‌ی اجرایی این الگوریتم $\mathcal{O}(n \log n)$ می‌باشد.

در مورد ضرورت داشتن الگوریتم SortNet می‌توان به این نکته اشاره کرد که اگر فرض کنیم به تعداد N عدد نمونه‌ی آزمایشی داشته باشیم، حال اگر رابطه‌ی ارجحیت هر جفت نمونه‌ی موجود در این دیتاست را داشته باشیم به تعداد $\binom{N}{2}$ عدد رابطه‌ی ارجحیت خواهیم داشت که آموزش این مقدار نیازمند زمان زیادی می باشد، لذا نیاز به الگوریتمی داریم که بتواند داده‌های ارجحیت را به صورت هدفمند برای آموزش شبکه‌ی CmpNN استفاده کند، که این عمل را الگوریتم SortNet انجام می‌دهد که در شکل ۴ آمده است. در این الگوریتم در هر دوره[۲۶] از آموزش، داده‌های آموزش توسط نمونه‌های جدید گسترش پیدا می‌کند. هدف از این کار این است که زیرمجموعه‌ای آموزشی از داده‌های آموزشی و تست انتخاب شود که دارای حداکثر بهره‌ی اطلاعاتی جهت بروزرسانی وزن‌های شبکه‌ی CmpNN باشند. هر مجموعه‌ی آموزشی برای آموزش شبکه‌ی CmpNN متفاوتی استفاده می‌شود. فرض کنید تابعی به نام RankQuality را داریم که میزان کیفیت یک رتبه‌بندی[۲۷] را ارزیابی می‌کند؛ نتیجه‌ی این الگوریتم تکراری شبکه‌ای است که بهترین نتیجه را بر اساس معیار ارزیابی داشته باشد. در الگوریتم SortNet مجموعه‌های P_V^i و P_T^i شامل جفت نمونه‌هایی ارزیابی و آموزشی هستند که با آن‌ها شبکه‌ی عصبی را آموزش می‌دهیم. در ابتدا این مجموعه‌های تهی هستند (P_T^0 و P_V^0 - خطوط ۳ و ۴) و شبکه به صورت تصادفی وزن‌دهی می‌شوند(خط ۵) و در ابتدا به شبکه ی خام(آموزش ندیده) داده‌های آموزشی و تست را می‌دهیم و نمونه‌هایی که توسط شبکه درست و غلط ارجحیت‌بندی شده اند را بدست می‌آوریم(خطوط ۱۰ و ۱۱) سپس برای آن‌هایی در مجموعه‌ی تست درست ارجحیت‌بندی شده‌اند را به تابع ارزیاب کیفیت می‌دهیم و امتیاز ارجحیت‌بندی مجموعه را بدست می‌آوریم(خط ۱۲) و در صورتی که این امتیاز از بهترین امتیاز بدست آمده تاکنون بیشتر باشد امتیاز و شبکه‌ی متناظر با آن را را به عنوان شبکه‌ی برگزیده ذخیره می‌کنیم(خطوط ۱۳ تا ۱۶). سپس داده‌های خطادار از مجموعه‌ی آموزشی و تست به ترتیب به داده‌هایی آموزشی و تستی که در دوره‌ی[۲۸] بعد که می‌خواهیم با آن‌ها شبکه را آموزشی دهیم و ارزیابی کنیم(خطوط ۷ تا ۹) اضافه می‌کنیم(خطوط ۱۷ و ۱۸) و در صورتی که شبکه خطایی نداشته باشد بهترین شبکه‌ی بدست آمده تاکنون را به عنوان خروجی برمی‌گردانیم(خطوط ۱۹ تا ۲۱). در صورتی که به حداکثر تعداد دوره رسیدیم بهترین شبکه‌ی بدست آمده را به عنوان خروجی برمی‌گردانیم(خط ۲۳).

۳-۳ ارزیابی کیفیت رتبه‌بندی

در این مقاله ۳ تابع ارزیابی کیفیت رتبه‌بندی(Ranking Quality) را معرفی کرده است که به شرح زیر می‌باشند:

۱. **P@n:** این معیار به صورت میزان ارتباط n مستند[۲۹] اول که رتبه‌بندی به آن‌ها ارجحیت بالاتری داده است را مشخص می‌کند؛ که به صورت زیر تعریف می‌شود:

$$\text{P@n} = \frac{\text{results n top in docs relevant}}{n} \qquad (۹)$$

۲. **MAP:** به عنوان میانگین‌گیری بروی معیار P@n می‌باشد:

$$\text{AP}_q = \frac{\sum_{n=1}^{N_q} \text{P@n} \cdot \text{rel}(n)}{N_q} \qquad (۱۰)$$

که در این‌جا N_q تعداد مستندات موجود در پرس‌وجو q[۳۰] می‌باشد. $\text{rel}(q)$ در زمانی که nامین مستند در مجموعه مرتبط با پرس‌وجو می‌باشد مقدار ۱ را اختیار می‌کند در غیر این صورت ۰ بر می‌گرداند.

۳. **NDCG@n:** به عنوان یک استخراج کننده‌ی صریح امتیازات رتبه‌ی بندی مجموعه مستندات معرفی شده است:

$$\text{NDCG@n} \equiv Z_n \left((2^{r_1} - 1) + \sum_{j=2}^{n} \frac{2^{r_j} - 1}{\log(j)} \right) \qquad (۱۱)$$

که در این‌جا $r_j \geq 0$ میزان ارتباط مستند jام با موضوع مورد پرس‌وجو می باشد(برای مستندات با ارتباط ضعیف این مقدار کوچک است) و Z_n یک فاکتور نرمال‌سازی می‌باشد بگونه‌ای که مقدار NDCG@n برای مرتب سازی ایده‌آل[۳۱] مقدار ۱ داشته باشد.

۳-۴ آزمایش‌ها

در آزمایشات از دیتاست‌هایی که توسط آزمایشگاه تحقیقاتی آسیا میکروسافت استفاده شده است که برای رتبه‌بندی و تعیین اولیت مستندات به ازای پرس‌وجوهای

مختلف استفاده می‌شوند. در این‌جا از نسخه‌ی ۲.۰ این مجموعه دیتاست استفاده شده است که شامل ۳ دیتاست زیر می‌باشد:

۱. **TD2003:** شامل ۵۰ مجموعه‌ی داده متناظر با ۵۰ پرس‌وجو که هریک شامل ۱۰۰۰ مستند می‌باشند. در این دیتاست هر زوج مرتب «پرس‌وجو، مستند» توسط ۴۴ مقدار که معمولا در بازیابی اطلاعات[۳۲] استفاده می‌شوند؛ نمایش داده‌می‌شوند. برچسب‌های تخصیص یافته به هر مستند نشان دهنده‌ی میزان ارتباط((Relevent (R)) و عدم‌ارتباط(Not Relevent (NR)) می‌باشد. به ازای هر پرس‌وجو فقط ٪۱ از مستندات مرتبط با موضوع آن می‌باشند.

۲. **TD2004:** همانند TD2003 می‌باشد با این تفاوت که شامل ۷۵ مجموعه‌ی داده‌ای تفاوت می‌باشد.

۳. **OHSUMED:** یک دیتاست مرتبط با کاربردهای پزشکی می‌باشد که شامل ۱۰۶ مجموعه‌ی مستندات پزشکی با میزان ارتباط موضوعی به ازای پرس‌وجوهای مختلف می‌باشد. در این دیتاست میزان ارتباط موضوعی هر پرس‌وجو با هر مستند توسط انسان تعیین شده‌اند که توسط ۳ مقدار ارتباط(R)، عدم‌ارتباط(NR) و احتمالامرتبط(PR) نشان داده شده‌اند. مستندات توسط ۲۵ ویژگی نمایش داده شده‌اند.

در ابتدای آزمایشات به بررسی و مقایسه‌ی علمکرد شبکه‌ی CmpNN با یکی از روش‌های دیگر بروی یادگیری تابع ارجحیت غیرمتعدی می‌پردازیم که در جدول ۲ آمده است. نتایج میانگینی از نتایج اجرای 5-Fold CrossValidation می‌باشد که در هر اجرا به‌صورت تصادفی ۱۰،۰۰۰ نمونه جهت آموزش و ۴،۰۰۰ نمونه جهت ارزیابی و ۶،۰۰۰ نمونه جهت تست استفاده شدند. آزمون آماری t تایید کرد که مدل ارائه شده در این مقاله بهبود قابل ملاحظه‌ای نسبت به روش دیگر داشته است. در جدول ۳ نیز عملکرد CmpNN با تغییر تعداد نورون‌های لایه‌ی مخفی بروی ۳ دیتاست TD2003، TD2004 و OHSUMED آورده شده است. در جدول ۴ نیز تعداد نورون‌های لایه‌ی مخفی برای دیتاست‌ها و توابع ارزیاب کیفیت مختلف آورده شده است که مبنای مقایسه‌های سایر نتایج آزمایشات دیگر این مقاله می‌باشد. در جداول ۵، ۶ و ۷ نتایج اجرای الگوریتم‌های مختلف به ترتیب بروی دیتاست‌های TD2003، TD2004 و OHSUMED که با معیارهای NDCG@n و P@n ارزیابی شده‌اند، نتایج نشان داده‌اند که الگوریتم معرفی شده(SortNet) توانسته است در هر ۳ دیتاست مورد آزمون، نتایج بهتری از دیگر الگوریتم‌های مدرن در این زمینه بدست آورد. در جدول ۸ نیز نتایج اجرای الگوریتم‌های مختلف بروی ۳ دیتاست معرفی شده که با معیار MAP سنجیده شده‌اند، آورده شده است، همان طور که می‌بینیم الگوریتم معرفی شده با این معیار مقایسه نیز توانسته بهتر از دیگر روش‌های مدرن عمل کند.

Hidden neurons		CmpNN	Neural Network of [20]
10	Test	**87.46 ± 1.67**	50.75 ± 29.11
	Train	**90.27 ± 1.41**	50.29 ± 29.53
	Validation	**87.09 ± 1.42**	49.56 ± 29.00
20	Test	**88.25 ± 0.56**	60.71 ± 12.15
	Train	**90.73 ± 0.39**	61.43 ± 12.06
	Validation	**88.02 ± 0.38**	60.02 ± 12.22
30	Test	**88.21 ± 0.95**	61.36 ± 19.27
	Train	**90.65 ± 1.78**	61.56 ± 20.24
	Validation	**87.63 ± 1.27**	62.02 ± 18.58
40	Test	**87.95 ± 0.74**	68.14 ± 7.15
	Train	**90.97 ± 1.85**	69.55 ± 7.68
	Validation	**87.93 ± 0.80**	69.14 ± 6.60
50	Test	**88.42 ± 1.08**	68.36 ± 12.01
	Train	**91.40 ± 2.06**	69.64 ± 11.95
	Validation	**88.07 ± 1.36**	68.88 ± 11.90

جدول۲: مقایسه‌ی عملکرد شبکه‌ی CmpNN با یک روش دیگر برای یادگیری تابع ارجحیت بروی داده‌های غیرمتعدی

Hiddens		TD2003	TD2004	OHSUMED
10	Accuracy	**92.93 ± 3.11**	97.22 ± 2.25	**94.38 ± 2.73**
	Precision	**92.95 ± 2.40**	97.15 ± 1.50	**94.93 ± 1.82**
	Recall	**92.95 ± 2.41**	97.15 ± 1.50	**94.93 ± 1.82**
20	Accuracy	90.66 ± 3.80	**97.97 ± 1.98**	94.28 ± 1.74
	Precision	90.11 ± 2.20	**97.64 ± 1.32**	94.11 ± 1.17
	Recall	90.10 ± 2.20	**97.65 ± 1.33**	94.19 ± 1.17
30	Accuracy	88.65 ± 3.94	96.63 ± 2.12	93.92 ± 1.45
	Precision	89.43 ± 2.29	97.08 ± 1.41	93.95 ± 0.96
	Recall	89.43 ± 2.29	97.08 ± 1.41	93.95 ± 0.97

جدول۳: مقایسه‌ی عملکرد شبکه‌ی CmpNN با تغییر تعداد نورون‌های لایه‌ی مخفی بروی دیتاست‌های مختلف

Dataset		Hidden neurons
TD2003	SortNet MAP	20
	SortNet P@10	30
	SortNet NDCG@10	20
TD2004	SortNet MAP	20
	SortNet P@10	10
	SortNet NDCG@10	10
OHSUMED	SortNet MAP	20
	SortNet P@10	10
	SortNet NDCG@10	10

جدول۴: تعداد نورون‌های لایه‌ی مخفی مورد استفاده در شبکه‌ی CmpNN در دیتاست‌های مختلف و توابع ارزیاب کیفیت مختلف

	NDCG@n										P@n									
	n=1	**n=2**	**n=3**	**n=4**	**n=5**	**n=6**	**n=7**	**n=8**	**n=9**	**n=10**	**n=1**	**n=2**	**n=3**	**n=4**	**n=5**	**n=6**	**n=7**	**n=8**	**n=9**	**n=10**
RankBoost	0.26	0.28	0.27	0.27	0.28	0.28	0.29	0.28	0.28	0.29	0.26	0.27	0.24	0.23	0.22	0.21	0.21	0.19	0.18	0.18
RankSVM	0.42	0.37	0.38	0.36	0.35	0.34	0.34	0.34	0.34	0.34	0.42	0.35	0.34	0.3	0.26	0.24	0.23	0.23	0.22	0.21
Frank-c19.0	0.44	0.39	0.37	0.34	0.33	0.33	0.33	0.33	0.34	0.34	0.44	0.37	0.32	0.26	0.23	0.22	0.21	0.21	0.2	0.19
ListNet	0.46	0.43	0.41	0.39	0.38	0.39	0.38	0.37	0.38	0.37	0.46	0.42	0.36	0.31	0.29	0.28	0.26	0.24	0.23	0.22
AdaRank MAP	0.42	0.32	0.29	0.27	0.24	0.23	0.22	0.21	0.2	0.19	0.42	0.31	0.27	0.23	0.19	0.16	0.14	0.13	0.11	0.1
AdaRank NDCG	0.52	0.41	0.37	0.35	0.33	0.31	0.3	0.29	0.28	0.27	0.52	0.4	0.35	0.31	0.27	0.24	0.21	0.19	0.17	0.16
SortNet MAP	**0.58**	**0.46**	**0.43**	**0.4**	**0.4**	**0.39**	**0.4**	**0.39**	**0.39**	**0.39**	**0.58**	**0.45**	**0.39**	**0.33**	**0.31**	**0.29**	**0.28**	**0.27**	**0.26**	**0.24**
SortNet P@10	**0.44**	**0.42**	**0.38**	**0.37**	**0.37**	**0.37**	**0.37**	**0.37**	**0.36**	**0.36**	**0.44**	**0.4**	**0.33**	**0.3**	**0.29**	**0.27**	**0.26**	**0.25**	**0.23**	**0.22**
SortNet NDCG@10	**0.5**	**0.42**	**0.41**	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**	**0.38**	**0.38**	**0.5**	**0.41**	**0.38**	**0.34**	**0.31**	**0.3**	**0.29**	**0.27**	**0.26**	**0.24**

جدول۵: نتایج اجرای الگوریتم‌های مختلف بروی دیتاست TD2003 براساس معیارهای NDCG@n و P@n

	NDCG@n										P@n									
	n=1	**n=2**	**n=3**	**n=4**	**n=5**	**n=6**	**n=7**	**n=8**	**n=9**	**n=10**	**n=1**	**n=2**	**n=3**	**n=4**	**n=5**	**n=6**	**n=7**	**n=8**	**n=9**	**n=10**
RankBoost	0.48	0.47	0.46	0.44	0.44	0.45	0.46	0.46	0.46	0.47	0.48	0.45	0.4	0.35	0.32	0.3	0.29	0.28	0.26	0.25
RankSVM	0.44	0.43	0.41	0.41	0.39	0.4	0.41	0.41	0.41	0.42	0.44	0.41	0.35	0.33	0.29	0.27	0.26	0.25	0.24	0.23
FRank	0.44	0.47	0.45	0.43	0.44	0.45	0.46	0.45	0.46	0.47	0.44	0.43	0.39	0.34	0.32	0.31	0.3	0.27	0.26	0.26
ListNet	0.44	0.43	0.44	0.42	0.42	0.42	0.43	0.45	0.46	0.46	0.44	0.41	0.4	0.36	0.33	0.31	0.3	0.29	0.28	0.26
AdaRank MAP	0.41	0.39	0.4	0.39	0.39	0.4	0.4	0.4	0.4	0.41	0.41	0.35	0.34	0.3	0.29	0.28	0.26	0.24	0.23	0.22
AdaRank NDCG	0.36	0.36	0.38	0.38	0.38	0.38	0.38	0.38	0.39	0.39	0.36	0.32	0.33	0.3	0.28	0.26	0.24	0.23	0.22	0.21
SortNet MAP	**0.47**	**0.47**	**0.46**	**0.46**	**0.45**	**0.45**	**0.45**	**0.46**	**0.47**	**0.48**	**0.47**	**0.43**	**0.38**	**0.36**	**0.33**	**0.3**	**0.28**	**0.28**	**0.27**	**0.26**
SortNet P@10	**0.39**	**0.43**	**0.42**	**0.44**	**0.44**	**0.45**	**0.46**	**0.46**	**0.46**	**0.47**	**0.39**	**0.4**	**0.36**	**0.36**	**0.33**	**0.32**	**0.31**	**0.28**	**0.27**	**0.26**
SortNet NDCG@10	**0.43**	**0.47**	**0.46**	**0.47**	**0.46**	**0.47**	**0.47**	**0.48**	**0.48**	**0.49**	**0.43**	**0.43**	**0.39**	**0.37**	**0.34**	**0.32**	**0.31**	**0.29**	**0.28**	**0.27**

جدول۶: نتایج اجرای الگوریتم‌های مختلف بروی دیتاست TD2004 براساس معیارهای NDCG@n و P@n

	NDCG@n										P@n									
	n=1	**n=2**	**n=3**	**n=4**	**n=5**	**n=6**	**n=7**	**n=8**	**n=9**	**n=10**	**n=1**	**n=2**	**n=3**	**n=4**	**n=5**	**n=6**	**n=7**	**n=8**	**n=9**	**n=10**
RankBoost	0.5	0.48	0.47	0.46	0.45	0.44	0.44	0.44	0.43	0.44	0.6	0.6	0.59	0.56	0.54	0.52	0.52	0.5	0.49	0.5
RankSVM	0.5	0.48	0.46	0.46	0.46	0.45	0.45	0.44	0.44	0.44	0.63	0.62	0.59	0.58	0.58	0.56	0.54	0.52	0.52	0.51
Frank-c4.2	0.54	0.51	0.5	0.48	0.47	0.46	0.45	0.45	0.44	0.44	0.67	0.62	0.62	0.58	0.56	0.53	0.51	0.5	0.5	0.49
ListNet	0.52	0.5	0.48	0.47	0.47	0.45	0.45	0.45	0.45	0.45	0.64	0.63	0.6	0.58	0.57	0.54	0.53	0.52	0.51	0.51
AdaRank.MAP	0.54	0.5	0.48	0.47	0.46	0.45	0.44	0.44	0.44	0.44	0.66	0.6	0.58	0.57	0.54	0.53	0.51	0.5	0.5	0.49
AdaRank.NDCG	0.51	0.47	0.46	0.46	0.44	0.44	0.44	0.44	0.44	0.44	0.63	0.6	0.57	0.56	0.53	0.53	0.52	0.51	0.5	0.49
MHR-BC	0.55	0.49	0.49	0.48	0.47	0.46	0.45	0.44	0.44	0.44	0.65	0.61	0.61	0.59	0.57	0.55	0.53	0.51	0.5	0.5
SortNet MAP	**0.52**	**0.48**	**0.47**	**0.46**	**0.45**	**0.45**	**0.44**	**0.44**	**0.44**	**0.44**	**0.63**	**0.58**	**0.57**	**0.56**	**0.55**	**0.55**	**0.53**	**0.52**	**0.51**	**0.5**
SortNet P@10	**0.49**	**0.43**	**0.44**	**0.43**	**0.42**	**0.42**	**0.42**	**0.42**	**0.42**	**0.42**	**0.62**	**0.57**	**0.57**	**0.55**	**0.53**	**0.51**	**0.51**	**0.51**	**0.5**	**0.49**
SortNet NDCG@10	**0.52**	**0.48**	**0.45**	**0.45**	**0.44**	**0.44**	**0.44**	**0.44**	**0.43**	**0.43**	**0.64**	**0.61**	**0.57**	**0.57**	**0.55**	**0.54**	**0.53**	**0.52**	**0.5**	**0.49**

جدول۷: نتایج اجرای الگوریتم‌های مختلف بروی دیتاست OHSUMED براساس معیارهای NDCG@n و P@n

	TD2003	TD2004	OSHUMED
RankBoost	0.212	0.384	0.44
RankSVM	0.256	0.35	0.447
Frank-c4.2	0.245	0.381	0.446
ListNet	0.273	0.372	0.45
AdaRank.MAP	0.137	0.331	0.442
AdaRank.NDCG	0.185	0.299	0.442
MHR-BC	NA	NA	0.44
SortNet MAP	**0.307**	**0.391**	**0.442**
SortNet P@10	**0.256**	**0.381**	**0.438**
SortNet NDCG@10	**0.297**	**0.402**	**0.44**

جدول۸: نتایج اجرای الگوریتم‌های مختلف بروی ۳ دیتاست مورد استفاده براساس معیار مقایسه‌ی MAP

۴ نتیجه‌گیری

در این گزارش ابتدا به بیان مفاهیم پایه‌ی یادگیری‌ارجحیت پرداختیم سپس با مروری خلاصه بر برخی کارهای انجام شده در زمینه‌ی یادگیری‌ارجحیت با تمرکز بروی مقاله‌ی اصلی سمینار به معرفی معماری جدیدی از شبکه‌ای عصبی که می تواند مدل ارجحیت بین داده‌ها را یادبگیرد پرداختیم سپس الگوریتمی که برای یادگیری بهینه‌ی شبکه استفاده می‌شود را شرح دادیم و در نهایت به ارائه نتایج حاصل از اجرای این شبکه و الگوریتم معرفی شده و مقایسه‌ی این نتایج با اجرای دیگر الگوریتم‌های مدرن در این زمینه پرداختیم و نشان دادیم که روش ارائه شده توانسته است با هر ۳ معیار ارزیابی کیفیت بهتر از دیگر روش عمل کند.

مراجع

[1] Rigutini, L., Papini, T., Maggini, M., Scarselli, F., *"SortNet: Learning to Rank by a Neural Preference Function,"* in Neural Networks, IEEE Transactions on , vol.22, no.9, pp.1368-1380, Sept. 2011

[2] Han-Jiang Lai, Yan Pan, Yong Tang, Rong Yu, *"FSMRank: Feature Selection Algorithm for Learning to Rank,"* in Neural Networks and Learning Systems, IEEE Transactions on , vol.24, no.6, pp.940-952, June 2013

[3] Fei Gao, Dacheng Tao, Xinbo Gao, Xuelong Li, *"Learning to Rank for Blind Image Quality Assessment,"* in Neural Networks and Learning Systems, IEEE Transactions on , vol.26, no.10, pp.2275-2290, Oct. 2015

[4] J. Furnkranz and E. Hullermeier, *"Encyclopedia of Machine Learning"*. Springer, 2010, ch. Preference Learning, pp. 789–795.

[5] J. Furnkranz and E. Hullermeier, *"Preference Learning"*. Kunstliche Intelligenz, pp. 60–61, 2005.

پانویس‌ها

[1] Rank
[2] Preference Learning
[3] Machnine Learning
[4] Order
[5] Preference Information
[6] Qualitative decision theory
[7] Non-monotonic reasoning
[8] Constraint satisfaction
[9] Planning
[10] Ranking Order
[11] Object Ranking
[12] Label Ranking
[13] Instance Ranking
[14] Total Strict Order
[15] Order
[16] Alternatives
[17] Format
[18] Index Permutation
[19] Classification
[20] Multi-label classification
[21] Learning Utility Functions
[22] Learning Preference Relations
[23] Feedforward
[24] Reflexivity
[25] Anti-symmetry
[26] Iteration
[27] Rank
[28] Iteration
[29] Document
[30] Query
[31] Ideal Ordering
[32] Information Retrieval