



پردیس علوم
دانشکده ریاضی، آمار و علوم کامپیوتر

آشنایی با الگوریتم‌های تکرار با فاصله: چگونه فلش‌کارت‌ها به مقابله با فراموشی کمک می‌کنند؟

نگارنده

محمد ترابی

استاد راهنما: دکتر هدیه ساجدی

پایان‌نامه برای دریافت درجه کارشناسی
در رشته علوم کامپیوتر

تاریخ: شهریور ۱۴۰۴

چکیده

این مقاله به کاوش در مکانیزم‌های بهبود حافظه از طریق فلش‌کارت‌ها و رویکرد **تکرار بافاصله** می‌پردازد. ابتدا، مفهوم **منحنی فراموشی** و پیامدهای آن برای یادگیری تشریح می‌شود. سپس، با معرفی ویژگی‌های یک فلش‌کارت مؤثر، به بررسی اصول شناختی پشت این ابزار پرداخته می‌شود. در بخش اصلی، **ویژگی‌های کلیدی الگوریتم‌های تکرار بافاصله** (مانند بهینه‌سازی فواصل و توزیع بار مرور) مورد بحث قرار گرفته و دو الگوریتم مهم یعنی Leitner و SuperMemo به تفصیل بررسی می‌شوند. در نهایت، با استناد به یافته‌های تحقیقاتی و آزمایشات وزنی‌اک، این پژوهش بر اهمیت طراحی الگوریتم‌های بهینه برای مدیریت مؤثر فرآیند یادگیری تاکید می‌کند.

سپاسگزاری

صمیمانه از استاد راهنمای گرانقدرم، سرکار خانم دکتر ساجدی، برای تدریس، راهنمایی‌ها و حمایت‌های ارزشمندشان سپاسگزارم. از جناب آقای دکتر باباعلی، مدیر محترم گروه، نیز بابت زحمات و آموزه‌های گرانقدرشان کمال تشکر را دارم. یاد و خاطره استاد ارجمند، مرحوم دکتر نوذری، که دانش و بینش ایشان همواره الهام‌بخش بود، گرامی باد.

پیشگفتار

یکی از چالش‌های بنیادین در فرآیند یادگیری، مقابله با پدیده طبیعی فراموشی است. از زمان پژوهش‌های پیشگامانه‌ای چون Hermann Ebbinghaus در حوزه **منحنی فراموشی**، همواره این سوال مطرح بوده است که چگونه می‌توان اطلاعات را به شکلی مؤثر و ماندگار در حافظه ذخیره کرد. این پژوهش، در تلاش برای ارائه پاسخی علمی به این سوال، به بررسی رویکردهای نوین در حوزه بهینه‌سازی فرآیند یادگیری می‌پردازد. در این راستا، **فلش‌کارت‌ها** به عنوان ابزاری قدرتمند و روش **تکرار بافاصله** به عنوان راهکاری علمی برای مبارزه با فراموشی، مورد توجه قرار گرفته‌اند. تلاش‌های نخستین در این زمینه با ابداع سیستم‌هایی چون روش Leitner آغاز شد و در ادامه، با توسعه الگوریتم‌های پیشرفته‌تری مانند SuperMemo توسط «Piotr Woźniak»، این حوزه وارد فاز جدیدی از دقت و کارایی گردید. هدف از این پژوهش، ارائه یک تحلیل جامع از اصول، مزایا و محدودیت‌های این روش‌هاست تا اهمیت رویکرد الگوریتمی در تقویت حافظه آشکار گردد. ساختار این پژوهش به گونه‌ای است که ابتدا مبانی نظری و علمی موضوع را تبیین می‌نماید. در ادامه، اصول طراحی فلش‌کارت‌های مؤثر و الگوریتم‌های کلیدی تکرار بافاصله مورد بررسی قرار می‌گیرد. همچنین نسخه نرم‌افزاری این سیستم‌ها به عنوان پیاده‌سازی عملی این پژوهش توسعه یافته است. کد منبع این پروژه و مستندات مربوط به آن را می‌توانید در مخزن گیت‌هاب مربوطه به آدرس github.com/mitinull/spaced-repetition-lab مشاهده کنید.

فهرست مطالب

۱	مفاهیم مقدماتی	۱
۱	۱.۱ فلش کارت	۱
۱	۲.۱ تکرار بافاصله ^۱	۱
۲	۳.۱ منحنی فراموشی ^۲	۲
۴	۲ اصول طراحی فلش کارت‌های کارآمد	۲
۴	۱.۲ ویژگی‌های یک فلش کارت خوب	۴
۵	۲.۲ مزایا و معایب مطالعه با فلش کارت‌ها	۵
۷	۳ اصول طراحی الگوریتم‌های تکرار بافاصله	۷
۷	۱.۳ ویژگی‌های کلیدی یک الگوریتم تکرار موثر	۷
۸	۲.۳ بازه‌های زمانی بهینه	۸
۱۰	۴ الگوریتم لایت‌نر ^۳	۱۰
۱۰	۱.۴ روش Leitner چگونه کار می‌کند؟	۱۰
۱۲	۲.۴ مزایای روش لایت‌نر	۱۲
۱۳	۳.۴ معایب روش لایت‌نر	۱۳
۱۴	۵ الگوریتم SuperMemo	۱۴
۱۴	۱.۵ روش SuperMemo چگونه کار می‌کند؟	۱۴
۱۴	۱.۱.۵ نحوه مرور و امتیازدهی به پاسخ‌ها	۱۴
۱۵	۲.۱.۵ محاسبه فاصله زمانی مرور بعدی	۱۵
۱۶	۳.۱.۵ ضریب سهولت	۱۶
۱۶	۴.۱.۵ الگوریتم SM-2	۱۶
۱۸	۲.۵ مزایای روش SuperMemo	۱۸

^۱ Spaced Repetition

^۲ Forgetting Curve

^۳ Leitner

۳.۵ معایب روش SuperMemo ۱۸

فصل ۱

مفاهیم مقدماتی

۱.۱ فلش کارت

فلش کارت‌ها ابزاری ساده اما بسیار **کارآمد** برای یادگیری و به‌خاطر سپاری اطلاعات هستند. این ابزارها، که می‌توانند به شکل کاغذی یا الکترونیکی باشند، شامل مجموعه‌ای از کارت‌ها هستند که روی یک یا هر دو طرف آن‌ها اطلاعاتی نوشته شده است. از فلش کارت‌ها می‌توان برای یادگیری طیف گسترده‌ای از موضوعات استفاده کرد، از جمله:

- **زبان‌های خارجی:** یادگیری واژگان جدید، مرور اشکال مختلف افعال بی‌قاعده
- **علوم پایه:** به‌خاطر سپردن مطالب در دروس پزشکی، شیمی، حقوق، تاریخ و جغرافیا
- **مهارت‌های تخصصی:** حفظ نت‌های گیتار، اصطلاحات علوم کامپیوتر
- **موارد روزمره:** مرور یادداشت‌های مهم، حفظ کردن شماره تلفن یا شماره کارت، به‌خاطر سپردن نام افراد از روی تصویرشان

۲.۱ تکرار بافاصله^۱

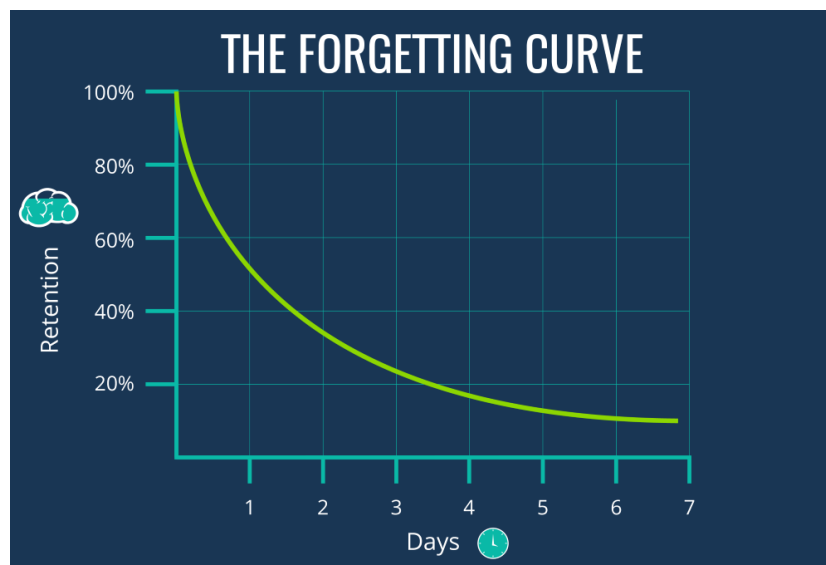
برای ماندگاری بیشتر اطلاعات در حافظه، مرور مطالب ضروری است. **تکرار بافاصله** روشی است که در آن مرور مطالب در فواصل زمانی مشخص و بهینه انجام می‌شود. این فواصل با گذشت زمان، به‌تدریج طولانی‌تر می‌شوند. این شیوه بر اساس الگوریتم‌هایی طراحی شده است که بهترین زمان را برای تکرار هر مطلب محاسبه می‌کنند. با استفاده از این روش، اثربخشی یادگیری به‌طور چشمگیری افزایش می‌یابد. [۳] در این مقاله چند الگوریتم مرتبط با تکرار بافاصله را بررسی خواهیم کرد.

^۱ Spaced Repetition

۳.۱ منحنی فراموشی^۲

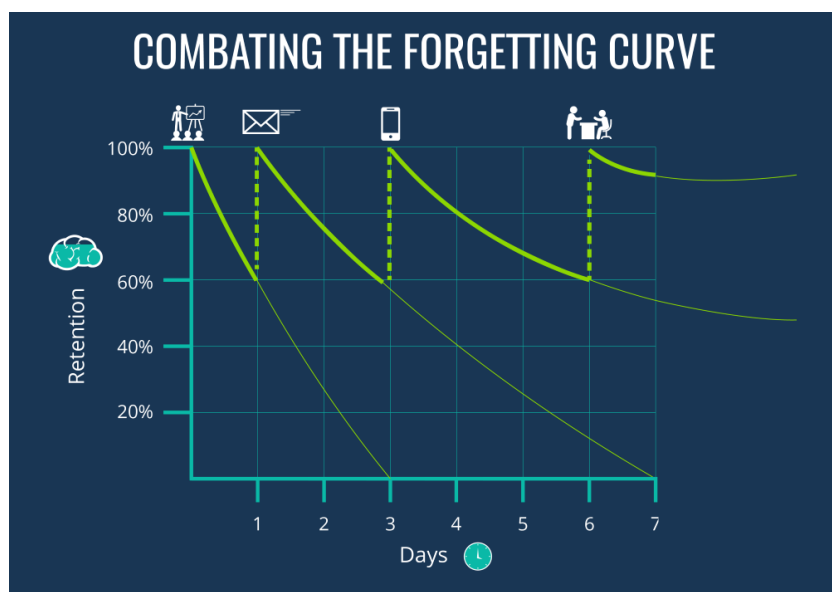
منحنی فراموشی نموداری است که نشان می‌دهد چگونه میزان به‌خاطرآوردن مطالب در طول زمان کاهش می‌یابد. این نمودار توسط روانشناس آلمانی، هرمان ابینگهاوس^۳، معرفی شد. محور افقی این نمودار نشان‌دهنده زمان و محور عمودی آن، **میزان یادآوری**^۴ یا احتمال به‌خاطرآوردن مطلب است.

همانطور که در شکل ۱.۱ مشخص است، اگر مطالب مرور نشوند، میزان یادآوری به‌سرعت کاهش می‌یابد. اما با استفاده از تکرار بافاصله، همانطور که در شکل ۲.۱ دیده می‌شود، شیب منحنی پس از هر بار مرور کمتر می‌شود. این بدان معناست که با هر بار تکرار، مدت‌زمان بیشتری طول می‌کشد تا مطلب فراموش شود و نیاز به مرور کمتری پیدا می‌کند.



شکل ۱.۱: منحنی فراموشی بدون مرور مطالب

^۲ Forgetting Curve
^۳ Hermann Ebbinghaus (1850–1909)
^۴ Retention



شکل ۲.۱: منحنی فراموشی با مرور مطالب

فصل ۲

اصول طراحی فلش کارت‌های کارآمد

۱.۲ ویژگی‌های یک فلش کارت خوب

طراحی به شکل پرسش و پاسخ:

بهترین فلش کارت‌ها از شما می‌خواهند به جای مرور صرف، به یک سوال پاسخ دهید. تحقیقات نشان داده‌اند که **یادآوری فعال**^۱، یعنی تلاش برای بازیابی اطلاعات، به یادگیری عمیق‌تر و ماندگارتر منجر می‌شود. این پدیده به «اثر آزمون»^۲ نیز معروف است.

کوتاه و مختصر:

یک فلش کارت باید شامل یک مفهوم یا واقعیت واحد باشد. مطالب کوتاه‌تر، یادگیری را ساده‌تر کرده و امکان مرور متناسب با میزان سختی هر بخش را فراهم می‌کنند. در مقابل، کارت‌های شلوغ مجبورمان می‌کنند کل محتوا را تکرار کنیم، حتی اگر بخشی از آن را بلد باشیم.

اجتناب از فهرست‌ها:

به جای پرسیدن «کشورهای خاورمیانه را نام ببرید؟»، بهتر است هر کشور را در یک کارت جداگانه با سوالات خاصی مانند «بزرگ‌ترین کشور خاورمیانه کدام است؟» یا «ثروتمندترین کشور آن کدام است؟» یاد بگیرید. این کار از ناکارآمدی حفظ فهرست‌های بلند جلوگیری می‌کند. سپس می‌توانید با پیوند دادن این اطلاعات، به سوال اصلی پاسخ دهید.

Active Recall^۱

Testing Effect^۲

منابع بیشتر:

برای آشنایی با ویژگی‌های دقیق‌تر و قوانین بهینه‌سازی فلش‌کارت‌ها، می‌توانید به مقاله «بیست قانون برای فرمول‌بندی دانش» [۲] مراجعه کنید. این مقاله توسط پاوو اولکوفسکی، بنیان‌گذار الگوریتم SuperMemo، نوشته شده است.

۲.۲ مزایا و معایب مطالعه با فلش‌کارت‌ها

مزایا:

- **افزایش ماندگاری:** با استفاده از فلش‌کارت‌ها، مطالب برای مدت‌زمان طولانی‌تری در حافظه می‌مانند. الگوریتم‌های تکرار بافاصله باعث می‌شوند مطالب در زمان بهینه مرور شوند و برای همیشه در حافظه باقی بمانند. این درحالی است که اکثر دانشجویان این فواصل بهینه را رعایت نمی‌کنند و حتی با مرور زیاد مطالب را پس از مدتی فراموش می‌کنند. [۴]
- **یادگیری فعال:** فلش‌کارت‌ها به دلیل ماهیت پرسش و پاسخ خود، یادگیری را فعال کرده و به جای مرور صرف، به یادآوری و بازیابی اطلاعات کمک می‌کنند.
- **مدیریت حجم بالای اطلاعات:** فلش‌کارت‌ها یادگیری محتوای گسترده را به بخش‌های کوچک‌تر و قابل‌مرور تقسیم می‌کنند. این ویژگی باعث می‌شود دانشجویان بدون احساس فشار، به تدریج بر حجم زیادی از مطالب تسلط پیدا کنند. خیلی از دانشجویان از تعداد کمی از فلش‌کارت‌ها استفاده می‌کنند که باعث می‌شود فواصل مرور کم شود و میزان یادگیری کاهش پیدا کند؛ درحالی که فلش‌کارت‌های بیشتر می‌توانند اثر یادگیری را افزایش دهند. [۴]
- **تمرکز بر نقاط ضعف:** با طبقه‌بندی کارت‌ها بر اساس میزان سختی، می‌توان روی مطالبی که تسلط کمتری بر آن‌ها دارید، بیشتر تمرکز کرد.
- **بازی‌سازی^۳:** افزایش جذابیت و انگیزه در یادگیری به دلیل ساده و بازی‌گونه بودن فلش‌کارت‌ها. همچنین آمارها، تحلیل‌ها و مصور سازی میزان پیشرفت مطالعه به ادامه دار شدن یادگیری کمک می‌کند.

معایب:

- **زمان‌بر بودن:** تهیه فلش‌کارت‌ها ممکن است زمان‌بر باشد، هرچند ابزارهای الکترونیکی این فرایند را ساده‌تر کرده‌اند.

^۳Gamification

- **نیاز به نظم و انضباط:** اثربخشی این روش به مرور منظم و مداوم وابسته است؛ این درحالی‌ست که پژوهش‌ها نشان می‌دهد دانشجویان غالباً بیشتر از پنج روز مطالعه را ادامه نمی‌دهند و هر بار بیش‌تر از نصف فلش‌کارت‌ها را نمی‌خوانند. [۶]

فصل ۳

اصول طراحی الگوریتم‌های تکرار بافاصله

۱.۳ ویژگی‌های کلیدی یک الگوریتم تکرار موثر

هدف بیشتر پژوهش‌ها در زمینه فلش‌کارت‌ها و تکرار بافاصله این است که الگوریتمی طراحی شود که بهترین تاثیر را در یادگیری داشته باشد. این هدف نیازمند داشتن ویژگی‌های متعددی است که جمع شدن همه آن‌ها در یک الگوریتم کار راحتی نیست. برخی از مهم‌ترین ویژگی‌های یک الگوریتم تکرار موثر در ادامه آمده است:

محاسبه بهینه فواصل تکرار:

هدف اصلی الگوریتم، یافتن بهترین زمان تکرار برای هر کارت است. این زمان باید طوری باشد که کارت درست قبل از اینکه فراموش شود، دوباره نمایش داده شود. این کار باعث می‌شود با کمترین تعداد مرور، اطلاعات برای بیشترین زمان ممکن در حافظه باقی بماند.

توزیع بار مرور:

یک الگوریتم هوشمند باید از انباشته شدن فلش‌کارت‌ها در یک روز خاص جلوگیری کند. به عبارت دیگر، وظیفه آن توزیع بهینه کارت‌ها در طول زمان است تا کاربر هر روز حجم معقول و مدیریت‌پذیری از کارت‌ها را برای مرور داشته باشد و از احساس خستگی یا عقب‌افتادگی جلوگیری شود. یعنی حجم فلش‌کارت‌ها در یک روز نباید خیلی کم و یا خیلی زیاد باشد.

تطبیق با سختی مطالب:

الگوریتم باید بر اساس عملکرد کاربر و سختی و آسانی هر کارت، فواصل تکرار را تنظیم کند. برای کارت‌های آسان‌تر، فاصله زمانی بیشتر می‌شود و برای کارت‌های دشوار، مرور در فواصل

کوتاه‌تری انجام می‌گیرد. این ویژگی، فرآیند یادگیری را شخصی‌سازی کرده و کارآمدتر می‌کند.

سازگاری با مطالعه نامنظم:

یک الگوریتم قوی باید با **مطالعه نامنظم** سازگار باشد و در صورت وقفه طولانی، دچار اختلال نشود. اگر کاربری برای چند روز یا هفته مطالعه نکند و سپس مرور را از سر بگیرد، الگوریتم باید این وقفه را درک کرده و فواصل زمانی را به‌درستی تنظیم کند. برای مثال، اگر قرار بود کارتی امروز مرور شود و سپس یک هفته بعد نمایش داده شود اما کاربر به جای امروز پس از یک ماه آن را مرور می‌کند، الگوریتم باید این فاصله زمانی طولانی را به عنوان یک «یادآوری موفق» ثبت کند و فاصله بعدی را بر اساس این واقعیت جدید، به جای یک هفته، به مراتب طولانی‌تر تعیین نماید.

۲.۳ بازه‌های زمانی بهینه

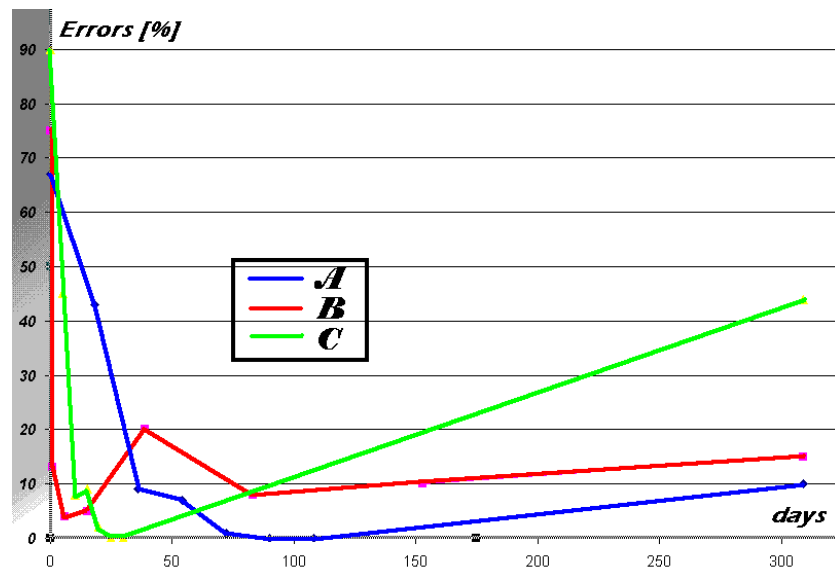
برخلاف تصور اولیه که هر فاصله‌ی زمانی افزایشی برای مرور مناسب است، «وزنیاک» با طراحی آزمایشی نشان داد که نتایج ممکن است متفاوت باشد. او برای بررسی این موضوع، تعدادی فعل بی‌قاعده‌ی انگلیسی را به سه گروه تقسیم کرد و هر گروه را شش بار با فواصل زمانی متفاوت مرور کرد. برنامه‌ی زمانی این آزمایش در جدول ۱.۳ آمده است.

جدول ۱.۳: برنامه زمانی مرور گروه‌های آزمایشی

نوبت مرور	گروه A	گروه B	گروه C
۱	روز ۱۸	روز ۱	روز ۵
۲	روز ۱۸	روز ۵	روز ۵
۳	روز ۱۸	روز ۹	روز ۵
۴	روز ۱۸	روز ۲۴	روز ۵
۵	روز ۱۸	روز ۴۴	روز ۵
۶	روز ۱۸	روز ۷۰	روز ۵
مجموع	روز ۱۰۸	روز ۱۵۳	روز ۳۰

قصد اصلی وزنیاک این بود که نشان دهد روش B (با فواصل افزایشی) بهترین نتیجه را به همراه دارد، اما نتایج که در شکل ۱.۳ مشاهده می‌کنید، متفاوت بود.

همانطور که در نمودار دیده می‌شود، گروه A با وجود خطای بیشتر در ابتدا، در درازمدت عملکردی پایدارتر داشت، در حالی که گروه C پس از یک دوره کوتاه بدون خطا، به سرعت دچار فراموشی شد. مهم‌تر از همه اینکه گروه A که با فواصل ثابت ۱۸ روز مرور می‌شد، در بلندمدت بهتر از گروه B با فواصل افزایشی عمل کرد. این نتیجه غافلگیرکننده، وزنیاک را به این جمع‌بندی



شکل ۱.۳: میانگین میزان خطای افراد در طول زمان در سه گروه آزمایشی

رساند که «هر فاصله‌ی افزایشی لزوماً مفید نیست» و یافتن الگویی بهینه برای زمان‌بندی تکرار، که از فواصل زمان ثابت هم کارآمدتر باشد، موضوع قابل توجهی است.

برای مطالعه بیشتر در مورد تحقیقات وزنیاک می‌توانید به فصل سوم مقاله [۱] و یا به لینک «شرح تحقیقاتی که منجر به روش SuperMemo شد: تابع تقریبی بازه‌های بهینه» مراجعه کنید.

فصل ۴

الگوریتم لایتنر^۱

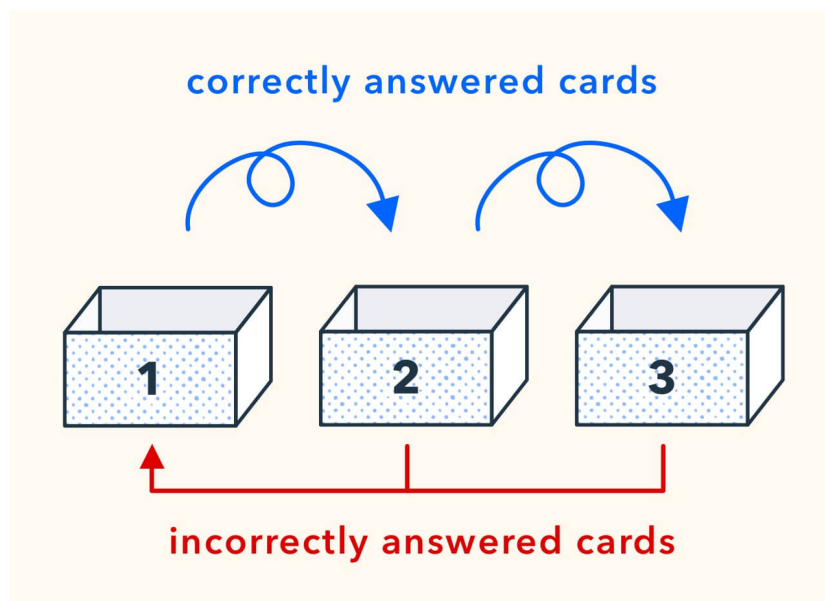
الگوریتم لایتنر، شناخته‌شده‌ترین الگوریتم در حوزه تکرار بافاصله است. این سیستم نوآورانه در دهه ۱۹۷۰ توسط **سباستین لایتنر**، روزنامه‌نگار و مروج علم آلمانی، توسعه یافت. او در کتاب تأثیرگذار خود با عنوان «این‌گونه باید آموخت»^۲ که در سال ۱۹۷۲ منتشر شد، این روش را به مخاطبان گسترده‌تری معرفی کرد. در ابتدا، این الگوریتم به صورت فیزیکی و با استفاده از جعبه‌هایی با چند خانه (معمولاً هفت خانه) پیاده‌سازی می‌شد. فلش‌کارت‌ها بر اساس میزان تسلط کاربر، بین این خانه‌ها جابجا می‌شدند. برای توضیحات بیشتر و یادگیری نحوه استفاده عملی از جعبه لایتنر به صورت فیزیکی، می‌توانید ویدیوی «چطور از جعبه لایتنر استفاده کنیم؟» را مشاهده کنید. این الگوریتم با وجود سادگی، نتایج بسیار قابل قبولی ارائه می‌دهد و یک روش عالی برای آشنایی و یادگیری اصول تکرار بافاصله است. به همین دلیل اول از همه به سراغ الگوریتم لایتنر می‌رویم.

۱.۴ روش Leitner چگونه کار می‌کند؟

روش لایتنر بر اساس تقسیم فلش‌کارت‌ها در خانه‌های مجزا است. این خانه‌ها بر اساس میزان تسلط شما بر مطالب، دسته‌بندی می‌شوند. خانه‌های اولیه برای مرور بیشتر و خانه‌های پایانی برای مرور کمتر در نظر گرفته شده‌اند. اصول کلی این روش به صورت زیر است:

- **حالت اولیه کارت‌ها:** همه فلش‌کارت‌ها در ابتدا در **خانه اول** قرار می‌گیرند.
- **جابه‌جایی کارت‌ها:** پس از هر بار مرور، اگر پاسختان صحیح باشد، کارت به خانه بعدی منتقل می‌شود. در غیر این صورت (پاسخ اشتباه)، کارت به **خانه اول** بازگردانده می‌شود.

^۱ Leitner
^۲ So lernt man lernen



شکل ۱.۴: شمای کلی از روش لایتنر

تا بیشتر مرور شود.

- **فواصل زمانی مرور:** این روش نسخه‌های مختلفی دارد، اما رایج‌ترین آن‌ها **جعبه‌ای با هفت خانه** است. در این نسخه، مرور کارت‌ها در هر خانه به صورت تصاعدی افزایش می‌یابد؛ به عنوان مثال، خانه اول هر روز، خانه دوم هر دو روز، خانه سوم هر چهار روز و... مرور می‌شود. [۷]

به عنوان مثال فرض کنید فلش‌کارتی را از خانه سوم بیرون آورده‌اید که روی آن کلمه "garcon" نوشته شده است، اما معنی آن را فراموش کرده‌اید. پس از مشاهده پاسخ (کلمه‌ای فرانسوی به معنی «پسر»)، کارت را به خانه اول برمی‌گردانید. **روز بعد**، چون خانه اول هر روز مرور می‌شود، دوباره "garcon" را می‌بینید. اگر این بار پاسخ صحیح بدهید، کارت به خانه دوم منتقل می‌شود. اگر پاسخ اشتباه بدهید، در همان خانه اول باقی می‌ماند.

روند مرور ادامه پیدا می‌کند تا زمانی که کارت به خانه آخر برسد و شما به طور کامل بر آن مسلط شوید. وقتی کارت از خانه آخر خارج می‌شود، به این معنی است که آن مطلب به طور کامل یاد گرفته شده است و تا مدت زمان زیادی آن را فراموش نمی‌کنیم. بنابراین می‌توانیم آن کارت را از روند مطالعاتی خارج کنیم.

شبه کد پایتونی و ساده شده الگوریتم لایتنر را می‌توانید در قطعه کد زیر مشاهده کنید.

1 DAY = 24 * 60 * 60 # seconds in a day (example)

```

2
3     def get_cards_to_review(cards):
4         """Return all cards that are due for review."""
5         review_cards = []
6
7         for card in cards:
8             if card.review_date <= current_date:
9                 review_cards.append(card)
10
11         return review_cards
12
13
14     def mark_correct(card):
15         """
16         Update the card if the user answered correctly.
17         - Increase the box number (spaced repetition level).
18         - Schedule next review further in the future.
19         """
20         card.box_number += 1
21         card.review_date = now() + (2 ** card.box_number) * DAY
22
23
24     def mark_incorrect(card):
25         """
26         Update the card if the user answered incorrectly.
27         - Reset box number.
28         - Schedule review for tomorrow.
29         """
30         card.box_number = 0
31         card.review_date = now() + 1 * DAY

```

۲.۴ مزایای روش لایتنر

روش لایتنر به طور خودکار و هوشمند نقاط ضعف شما را شناسایی و اولویت‌بندی می‌کند. کارت‌های دشوار در خانه‌های اولیه می‌مانند و بیشتر مرور می‌شوند، در حالی که مطالب آسان به خانه‌های آخر منتقل شده و کمتر تکرار می‌شوند. این سیستم با **منحنی فراموشی** تطابق بالایی دارد و سعی می‌کند که هر کارت قبل از فراموشی کامل، دوباره مرور شود و مدت زمان بیشتری در حافظه بماند. در نتیجه، این روش به طور بهینه زمان مطالعه را مدیریت کرده و از مطالعه بیش از حد مطالبی که از

قبل بر آن‌ها مسلط هستید، جلوگیری می‌کند.

۳.۴ معایب روش لایتنر

۱. **عدم توزیع هوشمندانه:** الگوریتم لایتنر، توزیع فلش‌کارت‌ها را بر اساس شروطی ساده انجام می‌دهد و همین موضوع می‌تواند منجر به نامنظم شدن حجم مطالعه در طول هفته شود. ممکن است در برخی روزها با حجم زیادی از کارت‌ها برای مرور مواجه شوید، در حالی که در روزهای دیگر، هیچ کارتی برای مطالعه نداشته باشید. برای مثال، اگر ۱۰ کارت جدید را در یک روز به سیستم اضافه کرده و به همه آن‌ها پاسخ درست دهید، این کارت‌ها به خانه دوم منتقل می‌شوند و دو روز بعد برای مرور مجدد ظاهر خواهند شد. در نتیجه، روز بعد هیچ کارتی برای مرور نخواهید داشت و این می‌تواند به برنامه‌ریزی شما لطمه بزند.

۲. **نادیده گرفتن سختی و آسانی پاسخ‌های درست:** این الگوریتم به میزان سختی یا آسانی یک پاسخ درست توجهی نمی‌کند. اگر مطلبی را به سختی به یاد آورید، آن را به خانه بعدی منتقل می‌کند؛ در حالی که اگر به مطلبی کاملاً مسلط باشید و به راحتی آن را به یاد آورید، باز هم همین کار را می‌کند. این عدم تفاوت‌گذاری باعث می‌شود که کارت‌های دشوار و آسان به یک شیوه مدیریت شوند و این می‌تواند کارایی سیستم را کاهش دهد و باعث شود بعضی کارت‌ها بیش از حد مرور شوند.

۳. **ضعف در تطبیق با وقفه‌های مطالعه:** روش لایتنر با وقفه‌های طولانی در مطالعه سازگاری خوبی ندارد. اگر برای مثال، یک کارت را در خانه اول قرار داده و پس از یک ماه به سراغ آن برگردید و جواب آن را بلد باشید، سیستم همچنان آن را به خانه دوم منتقل کرده و دو روز بعد به شما نشان می‌دهد. این در حالی است که وقتی بعد از یک ماه هنوز مطلب را به یاد دارید، نشان‌دهنده تسلط شما بر آن است و نیازی به مرور مجدد در فاصله زمانی کوتاه نیست. این نقص در بهینه‌سازی، زمان و انرژی شما را هدر می‌دهد.

فصل ۵

الگوریتم SuperMemo

سوپر ممو یکی از پیشرفته‌ترین و پرکاربردترین الگوریتم‌های تکرار بافاصله است که توسط **پیوتر وُزنیاک**^۱، پژوهشگر لهستانی، توسعه یافت. این الگوریتم به مرور زمان نسخه‌های مختلفی پیدا کرده (از SM-0 تا SM-18)، اما نسخه‌ی معروف و پرکاربرد آن SM-2 است که هنوز هم اساس بسیاری از نرم‌افزارهای یادگیری مثل Anki^۲ را تشکیل می‌دهد. [۸] همچنین یک نرم‌افزار به نام SuperMemo وجود دارد که که توسط خود وُزنیاک توسعه یافته است و از نسخه‌های بالاتر این الگوریتم استفاده می‌کند.

۱.۵ روش SuperMemo چگونه کار می‌کند؟

این روش پیچیدگی‌های بیشتری نسبت به روش لایت‌ر دارد و از چندین پارامتر برای تنظیم فواصل مرور استفاده می‌کند. همچنین قواعد مرور نیز تفاوت‌هایی با روش لایت‌ر دارد.

۱.۱.۵ نحوه مرور و امتیازدهی به پاسخ‌ها

هنگامی که یک فلش‌کارت را با روش SuperMemo مرور می‌کنید، الگوریتم از شما می‌خواهد که عملکرد خود را بر اساس یک مقیاس درجه‌بندی (از ۰ تا ۵) ارزیابی کنید. وُزنیاک این مقیاس را کیفیت پاسخ^۳ یا به اختصار q نامیده است. این امتیازدهی به الگوریتم کمک می‌کند تا بفهمد چقدر بر مطلب مسلط هستید و سپس با استفاده از فرمول‌های پیچیده‌تر، فاصله زمانی بعدی را محاسبه می‌کند. در ادامه می‌توانید ببینید که هر کدام از این امتیازها به چه معنیست:

^۱Wozniak Piotr

^۲Anki یک نرم‌افزار معروف برای مطالعه فلش‌کارت با استفاده از الگوریتم‌های با فاصله است.
^۳quality of response

- ۵ - پاسخ کامل و بدون مشکل (perfect response)
- ۴ - پاسخ صحیح با کمی تردید (correct response after a hesitation)
- ۳ - پاسخ صحیح همراه با سختی زیاد (correct response recalled with serious difficulty)
- ۲ - پاسخ اشتباه؛ اما یادآوری پاسخ صحیح آسان بود (incorrect response; where the correct one seemed easy to recall)
- ۱ - پاسخ اشتباه؛ اما پاسخ صحیح به یاد آورده شد (incorrect response; the correct one remembered)
- ۰ - فراموشی کامل (complete blackout)

۲.۱.۵ محاسبه فاصله زمانی مرور بعدی

در روش SuperMemo فاصله زمانی با استفاده از فرمول بازگشتی زیر محاسبه می‌شود.

$$I(0) = 0 \quad (۱.۵)$$

$$I(1) = 1 \quad (۲.۵)$$

$$I(2) = 6 \quad (۳.۵)$$

$$I(n) = I(n-1) \times EF \quad (۴.۵)$$

با توجه به فرمول محاسبه فاصله زمانی، می‌توانیم ببینیم که **فاصله زمانی برای مرور اول** برابر با ۱ روز و **برای مرور دوم** برابر با ۶ روز است. از اینجا به بعد، فاصله زمانی با توجه به ضریب **سهولت**^۴ (EF) کارت محاسبه می‌شود. به عنوان مثال، اگر ضریب سهولت یک کارت برابر با ۲ باشد، فاصله زمانی برای مرور سوم برابر با ۱۲ روز خواهد بود (۶ ضربدر ۲). اگر ضریب سهولت برابر با ۲/۵ باشد، فاصله زمانی برای مرور سوم برابر با ۱۵ روز خواهد بود (۶ ضربدر ۲/۵). در نتیجه مرور اول و دوم برای همه کارت‌ها یکسان است، اما از مرور سوم به بعد، هرچه کارت آسان‌تر باشد، فاصله زمانی بیشتری برای مرور بعدی آن تعیین می‌شود. همانطور که می‌بینیم کیفیت پاسخ (q) تاثیر مستقیمی بر بازه زمانی بعدی ندارد. تاثیر این فاکتور از طریق ضریب سهولت به صورت غیرمستقیم اعمال می‌شود. در قسمت ۴.۱.۵ خواهیم دید که اگر q کمتر از ۳ باشند، r به ۱ بازنشانی می‌شود، در نتیجه فاصله زمانی به ۱ روز کاهش می‌یابد.

Ease Factor^۴

۳.۱.۵ ضریب سهولت

هر کارت یک ضریب مخصوص دارد که نشان دهنده‌ی میزان آسانی یا دشواری آن است. اگر کارتی برای شما دشوار باشد و اغلب نمره‌ی پایینی بدهید، E-Factor آن کاهش می‌یابد و مرورهای آینده نزدیک‌تر می‌شوند. اگر کارت آسان باشد، E-Factor آن بالا می‌رود و مرورها دورتر می‌شوند. ضریب سهولت در ابتدا برابر با ۲/۵ تنظیم می‌شود و پس از هر بار مرور، این ضریب با استفاده از فرمول زیر محاسبه می‌شود:

$$EF' = EF + (0.1 - (5 - q) \times (0.08 + (5 - q) \times 0.02)) \quad (۵.۵)$$

که شکل کاهش یافته‌ی آن به صورت زیر است:

$$EF' = EF - 0.8 + 0.28q - 0.02q^2 \quad (۶.۵)$$

توجه کنید که اگر کیفیت پاسخ برابر ۴ باشد، ضریب سهولت تغییری نمی‌کند. همچنین طبق پژوهش‌های وزنیاک، ضریب سهولت نباید کمتر از ۱/۳ باشد؛ زیرا در این صورت تعداد مرورها به شکل آزاردهنده‌ای افزایش پیدا می‌کند. بنابراین اگر محاسبات منجر به ضریب سهولت کمتر از ۱/۳ شود، آن را برابر با ۱/۳ در نظر می‌گیریم.

۴.۱.۵ الگوریتم SM-2

حال که با نحوه‌ی محاسبه فاصله زمانی و ضریب سهولت آشنا شدیم، می‌توانیم نسخه‌ی کامل الگوریتم SM-2 که یکی از معروف‌ترین نسخه‌های SuperMemo است را بررسی کنیم. الگوریتم به این صورت عمل می‌کند:

۱. در ابتدا برای هر کارت ضریب سهولت را ۲/۵، تعداد مرور (r) را ۱ و آخرین بازه زمانی را برابر با ۰ قرار می‌دهیم.

۲. وقتی نوبت به مرور کارت رسید از کاربر می‌خواهیم که کیفیت پاسخ (q) را از ۰ تا ۵ مشخص کند. سپس با توجه به مقدار q، مراحل زیر را انجام می‌دهیم:

(آ) اگر کیفیت پاسخ (q) کمتر از ۳ باشد:

- تعداد مرور (r) را برابر به ۱ بازنشانی می‌کنیم.
- زمان مرور بعدی را با توجه به فرمول ۲.۵ یک روز بعد قرار می‌دهیم.
- اگر کیفیت پاسخ (q) برابر یا بیشتر از ۳ باشد:
- زمان مرور بعدی را با توجه به فرمول‌های ۲.۵ تا ۴.۵ محاسبه می‌کنیم.
- تعداد مرور (r) را یک واحد افزایش می‌دهیم.

(ب) ضریب سهولت را با توجه به فرمول ۶.۵ به روزرسانی می‌کنیم و اگر کمتر از $\frac{1}{3}$ شد، آن را برابر با $\frac{1}{3}$ قرار می‌دهیم.

شبه کد پایتونی و ساده شده الگوریتم SM-2 را می‌توانید در قطعه کد زیر مشاهده کنید.

```
1 DAY = 24 * 60 * 60 # seconds in a day
2
3 def get_cards_to_review(cards):
4     """Return all cards that are due for review."""
5     review_cards = []
6
7     for card in cards:
8         if card.review_date <= current_date:
9             review_cards.append(card)
10
11     return review_cards
12
13
14 def mark_card(card, q):
15     """
16     Update the card based on the user's quality of response (q).
17     q is an integer from 0 to 5.
18     """
19     if q < 3:
20         # If the answer was incorrect or hard to recall
21         card.repetition = 1
22         card.review_date = now() + 1 * DAY
23     else:
24         # If the answer was correct
25         if card.repetition == 1:
26             interval = 1
27         elif card.repetition == 2:
28             interval = 6
29         else:
30             interval = card.interval * card.ease_factor
31
32         card.repetition += 1
33         card.review_date = now() + interval * DAY
34
35     # Update ease factor
```

```

36         card.ease_factor += (0.1 - (5 - q) * (0.08 + (5 - q) * 0.02))
37         if card.ease_factor < 1.3:
38             card.ease_factor = 1.3

```

پیاده‌سازی عملی و کامل این الگوریتم را می‌توانید در مخزن گیت‌هاب مربوط به این پروژه مشاهده کنید.

۲.۵ مزایای روش SuperMemo

این الگوریتم نسخه پیشرفته‌تری از روش لایتنر است و بیشتر مزایای روش لایتنر را دارد. علاوه بر این، با استفاده از امتیازدهی به پاسخ‌ها، می‌تواند فواصل زمانی را به صورت دقیق‌تری تنظیم کند و مشکل در نظر نگرفتن میزان سختی در پاسخ‌ها برطرف می‌شود. این ویژگی باعث می‌شود که زمان مرور به نحو احسن مدیریت شود و مطالعه بیش از حد از مطالب آسان کاهش یابد. همچنین تنوع در فواصل زمانی مرور باعث می‌شود که حجم مطالعه در طول هفته به شکل بهتری توزیع شود و از انباشتگی کارت‌ها در یک روز خاص جلوگیری شود.

۳.۵ معایب روش SuperMemo

اگر چه این الگوریتم نسبت به روش لایتنر پیشرفته‌تر است، اما همچنان معایبی دارد که برطرف کردن آن‌ها نیازمند الگوریتم‌های پیچیده‌تر است و همچنان محققان مشغول به پژوهش در این زمینه هستند. از جمله معایب این روش می‌توان به موارد زیر اشاره کرد:

۱. **پیچیدگی امتیازدهی:** الگوریتم SuperMemo نیازمند امتیازدهی دقیق از سوی کاربر است. این ممکن است برای برخی افراد دشوار باشد و باعث شود که امتیازها به صورت نادرست ثبت شوند.

۲. **عدم تطبیق با وقفه‌های مطالعه:** همانند روش لایتنر، این الگوریتم نیز با وقفه‌های طولانی در مطالعه سازگاری خوبی ندارد. اگر برای مثال، یک کارت را در فاصله زمانی کوتاه مرور کرده و سپس برای مدت طولانی به سراغ آن نروید، ممکن است الگوریتم نتواند به درستی تشخیص دهد که شما بر آن مطلب مسلط هستید یا خیر.

۳. **عدم شخصی سازی:** اگر چه این الگوریتم از امتیازدهی برای تنظیم فواصل زمانی استفاده می‌کند، اما همچنان از یک فرمول ثابت برای همه کاربران بهره می‌برد. این ممکن است باعث شود که الگوریتم نتواند به خوبی با سبک یادگیری و نیازهای خاص هر فرد سازگار شود. [۵]

۴. **عدم توجه به محتوی و نوع کارت:** این الگوریتم به محتوای کارت‌ها و نوع آن‌ها توجهی ندارد. برای مثال، یادگیری یک زبان جدید ممکن است نیازمند فواصل زمانی متفاوتی نسبت به یادگیری مفاهیم علمی باشد. این الگوریتم نمی‌تواند این تفاوت‌ها را در نظر بگیرد و ممکن است برای همه نوع مطالب به یک شکل عمل کند. [۶]

نتیجه‌گیری

این پژوهش با هدف مقابله با چالش بنیادین فراموشی، به بررسی عمیق اصول و الگوریتم‌های **تکرار بافاصله** پرداخت. یافته‌ها به وضوح نشان می‌دهند که فلش‌کارت‌ها، بیش از یک ابزار ساده برای حفظ کردن هستند؛ آن‌ها ابزاری قدرتمند برای **یادگیری فعال** و بازیابی اطلاعات به شمار می‌روند. این روش، با استفاده از فواصل مرور هوشمندانه، به‌طور چشمگیری از اثرات **منحنی فراموشی** می‌کاهد و مطالب را از حافظه کوتاه‌مدت به حافظه بلندمدت منتقل می‌کند. همان‌طور که مشاهده شد، الگوریتم‌های مختلفی برای پیاده‌سازی این سیستم‌ها وجود دارد؛ از روش ساده اما کارآمد Leitner که اساس آن بر جابه‌جایی کارت‌ها در خانه‌های مجزا است، تا الگوریتم‌های پیشرفته‌تری چون SuperMemo که با استفاده از داده‌های عملکردی کاربر، فواصل زمانی را به صورت پویا و کاملاً شخصی‌سازی شده تنظیم می‌کنند. در این میان، نتایج آزمایشات «وزنیاک» به ما آموخت که هر نوع تکرار بافاصله‌ای لزوماً کارآمد نیست و کلید موفقیت در یافتن الگوی بهینه و هوشمندانه برای زمان‌بندی مرور است. در مجموع، این پژوهش تأکید می‌کند که بهینه‌سازی فرآیند یادگیری فراتر از مطالعه صرف است و در گرو انتخاب ابزارها و الگوریتم‌های مناسب قرار دارد. استفاده آگاهانه از فلش‌کارت‌ها و **الگوریتم‌های هوشمند** تکرار بافاصله، نه تنها زمان مطالعه را مدیریت می‌کند، بلکه به کاربر امکان می‌دهد تا بر نقاط ضعف خود تمرکز کرده و با کمترین تلاش، به بیشترین ماندگاری اطلاعات دست یابد. این رویکرد، در نهایت به تسلطی عمیق‌تر و پایدارتر بر دانش منجر خواهد شد.

کتابنامه

- [1] Woźniak, Piotr, and Edward Gorzelańczyk. "Optimization of repetition spacing in the practice of learning." *Acta neurobiologiae experimentalis* 54.1 (1994): 59-62.
- [2] P. Wozniak, "Effective learning: Twenty rules of formulating knowledge", <https://super-memory.com/articles/20rules.htm>, 1999.
- [3] Hunshamar, Asgeir. A Flashcard Based Web Application for Collective Learning and Peer Review Based Evaluation of Students. MS thesis. NTNU, 2021.
- [4] Wissman, Kathryn T., Katherine A. Rawson, and Mary A. Pyc. "How and when do students use flashcards?." *Memory* 20.6 (2012): 568-579.
- [5] Ye, Junyao, Jingyong Su, and Yilong Cao. "A stochastic shortest path algorithm for optimizing spaced repetition scheduling." *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 2022.
- [6] Shu, Matthew, et al. "Karl: Knowledge-aware retrieval and representations aid retention and learning in students." *arXiv preprint arXiv:2402.12291* (2024).
- [7] Settles, Burr, and Brendan Meeder. "A trainable spaced repetition model for language learning." *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*. 2016.
- [8] Pokrywka, Jakub, et al. "Modeling Spaced Repetition with LSTMs." *CSEDU* (2). 2023.

Abstract

This paper explores the mechanisms of memory enhancement through flashcards and the **spaced repetition** approach. First, the concept of the **forgetting curve** and its implications for learning are explained. Then, by introducing the characteristics of an effective flashcard, the cognitive principles behind this tool are examined. In the main section, the **key features of spaced repetition algorithms** (such as optimizing intervals and distributing review load) are discussed, and two important algorithms, **Leitner** and **SuperMemo**, are examined in detail. Finally, based on research findings and the experiments of **Wozniak**, this study emphasizes the importance of designing optimal algorithms for the effective management of the learning process.



College of Science
School of Mathematics, Statistics, and Computer Science

An Introduction to Spaced Repetition Algorithms: How Flashcards Help Combat Forgetting?

Mohamamd Torabi

Supervisor: Dr. Hedieh Sajedi

A thesis submitted in partial fulfillment of the requirements for
the degree of B.Sc. in Computer Science

2025