فصل اول: مقدمه

از زمانی که رایانه‌ها ساخته شده‌اند، انسان‌ها همواره به دنبال راهی بوده تا بتوانند آن‌ها را برای مقاصد دلخواهشان، آموزش دهند تا شاید بتوانند روزی آن‌ها را طوری برنامه‌ریزی کنند که بتوانند خودشان با گذر از آزمایش‌ها، بر تجربه‌ی خود بیفزایند و هوشمند شوند. می‌توان روزی را تصور کرد که رایانه‌ها می‌توانند از روی‌داده‌های درمانی نحوه‌ی تشخیص بیماری و روش درمان مؤثرتر را پیدا کنند؛ در ساختمان‌ها در اثر گذشت زمان و با در نظر گرفتن داده‌های انرژی، بهینه‌ترین برنامه‌ی انرژی را برای ساختمان تنظیم کنند؛ در نرم‌افزارهای شخصی، با توجه به سلیقه‌تان، برنامه‌ی مورد نظر را برایتان پیشنهاد دهند. در واقع با موفقیت در آموزش صحیح به رایانه‌ها، دروازه‌های جدیدی از زندگی برای انسان‌ها، باز خواهد شد. همچنین پیشرفت بهتری در زمینه‌ی الگوریتم‌های تجزیه و تحلیل اطلاعات، به ما کمک خواهد کرد تا توانایی‌های انسان (یا حتی محدودیت‌های آن را!) بهتر دریابیم.

در حال حاضر، ما دقیقاً نمی‌دانیم چگونه باید رایانه‌ها را برنامه‌ریزی کنیم تا به خوبی انسان‌ها یاد بگیرند. هر چند که روش‌هایی که تاکنون کشف شده‌اند، برای اهدافی خاص، بسیار مؤثر عمل می‌کنند اما برای تمامی اهداف مناسب نیستند. برای مثال در کاوش اطلاعات[۱]، استفاده از الگوریتم‌های یاددهی به ماشین، بسیار متداول است. حتی در زمینه‌هایی با داده‌ها سر و کار دارند، این الگوریتم‌ها بسیار بیش از حد انتظار عمل کرده و جواب داده‌اند. به عنوان مثال در مسائلی مانند شناسایی گفتار[۲]، الگوریتم‌های مبتنی بر یادگیری ماشین، بسیار بهتر از سایر روش‌ها، جواب داده‌اند. ظاهراً به نظر می‌رسد دانش ما از رایانه‌ها، رفته رفته، به بلوغ می‌رسد. به جرأت می‌توان گفت، مبحث یاددهی به ماشین، نقشی به شدت پررنگ در زمینه‌ی علوم کامپیوتر و تکنولوژی کامپیوتری بازی می‌کند.

دستاوردهایی در این زمینه به دست آمده است: برنامه‌هایی نوشته شده‌اند که یاد می‌گیرند که صدای کلمات را تشخیص دهند (Waibel 1989; Lee 1989)، ضریب بهبود بیماران ذات‌الریه را پیش‌بینی کنند (Cooper 1997)، کلاه‌برداری با کارت اعتباری را تشخیص دهند،

---

[۱] Data Mining

[۲] Speech Recognition

اتومبیل‌ها را در بزرگراه هدایت کنند (Pomerleau 1989) و بازی‌هایی مثل تخت نرد[۱] را در حد انسان‌های ماهر بازی کنند (Tesauro 1992,1995). نتیجه‌های تئوری‌ای به دست آمده که روابط پایه‌ای بین تعداد نمونه‌های آموزشی مشاهده شده و تعداد فرضیه‌های ممکن و امید میزان خطا در فرضیه‌ها را مشخص می‌کنند. آدمی در عصر حاضر کم کم به مدل‌های اولیه‌ی یادگیری انسان و حیوان پی می‌برد و کم کم رابطه‌ی الگوریتم‌های یادگیری کامپیوتری را با این مدل‌ها پیدا می‌کند (Lair 1986; Anderson 1991; Qin 1992; Chi and Bassock 1989; Ahn and Brewer 1993). در عمل نیز، در دهه‌های اخیر الگوریتم‌ها، تئوری و تحقیقات بر روی سیستم‌های زیستی یادگیری پیشرفت قابل توجهی کرده‌اند. خلاصه‌ی تعداد بسیاری از پروژه‌های یادگیری ماشین در جدول ۱,۱ آمده است. Langley and Simon (1995) و Rumelhart (1994) کاربردهای دیگری را در یادگیری ماشین تحقیق کردند.

لذا در این نوشته ما سعی خواهیم کرد، مباحث ، الگوریتم‌های یادگیری ، نتایج نظری و کاربردهای آن‌ها را مورد بررسی قرار دهیم. به دلیل ویژگی ذاتی این مبحث در ارتباط آن با رشته‌ها و زمینه‌های گوناگون، مانند مباحث هوش مصنوعی، آمار و احتمالات، هندسه محاسباتی، تئوری کنترل،تئوری اطلاعات، فلسفه، روان‌شناسی، عصب‌شناسی و ... ، هر جا که لازم باشد، مباحث را در حد نیاز بررسی خواهیم کرد. جدول ۱,۲ ایده‌های اصلی که یاددهی به ماشین با علوم مختلف دیگر دارد را به صورت خلاصه بیان کرده است. از آنجایی که هدف از این کتاب بکار گیری نتیجه‌های به دست آمده از این تحقیقات است، لازم نیست خواننده در این زمینه‌ها حرفه‌ای باشد. نکته‌های کلیدی این زمینه‌ها معمولاً با زبانی ساده بیان شده است و جملات و عبارات ناآشنا نیز تعریف خواهند شد.

۱,۱ مسائل یادگیری خوش وضع[۲]

بیایید مطالعه‌ی یادگیری ماشین را با معرفی چند عمل یادگیری شروع کنیم. ابتدا مفهوم یادگیری[۳] را به فرمی تعریف می‌کنیم که هرگونه برنامه کامپیوتری که کارایی‌اش در کار خاصی با تجربه بهبود یابد را در بر گیرد. به عبارت دقیق‌تر،

تعریف: زمانی گفته می‌شود که یک برنامه‌ی کامپیوتری از تجربه‌ی[۴] E در مورد کار[۵] T بر حسب معیار کارایی[۶] P یادگیری دارد که کارایی‌اش بعد از تجربه‌ی E برای کار T بهبود بیابد.

برای مثال، برنامه‌ی کامپیوتری‌ای که یاد می‌گیرد تا چکرز[۷] بازی کند می‌تواند کارایی خود را که با "توانایی بردن" معلوم می‌گردد، اعمال ممکن بازی چکرز را اعمال ممکن و تجربه‌ای که از بازی در مقابل خودش به دست می‌آورد را تجربه در نظر گرفت. در کل، برای اینکه مسئله، مسئله‌ای خوش وضع باشد، باید ویژگی‌های روبرو را برای آن معلوم کنیم: مجموعه‌ی اعمال ممکن، کارایی‌ای که باید بهبود یابد و منبع تجربیات.

---

[۱] Backgammon
[۲] well-posed
[۳] learning
[۴] experience
[۵] task
[۶] performance
[۷] checkers

مسئله‌ی یادگیری بازی چکرز:

- عمل T: بازی کردن چکرز.
- کارایی P: درصد بازی‌های برده در مقابل حریف.
- تجربیات آموزشی E: بازی تمرینی در مقابل خودش.

به همین منوال می‌توان مسئله‌های یادگیری خوش وضع بسیاری را نظیر مسئله‌هایی چون یادگیری تشخیص دستخط[۱] و یا یادگیری هدایت یک اتومبیل مشخص کرد.

مسئله‌ی یادگیری تشخیص دستخط:

- عمل T: تشخیص و دسته‌بندی کلمات دست‌نویس در تصاویر
- کارایی P: درصد کلماتی که درست دسته‌بندی شده‌اند
- تجربیات آموزشی E: پایگاه داده‌ای از کلمات دست‌نویس با دسته‌بندی‌هایشان.

مسئله‌ی یادگیری هدایت یک اتومبیل:

- عمل T: هدایت اتومبیل در آزادراه با استفاده از دوربین‌های نصب شده
- کارایی P: میزان طولی که بدون خطا اتومبیل هدایت شده (خطا ممکن است توسط عامل انسانی تشخیص داده شود)
- تجربیات آموزشی E: مجموعه‌ای از دستورهای هدایت و عکس‌های مربوطه‌ی دوربین‌ها در زمان هدایت اتومبیل توسط انسان

- یادگیری تشخیص صوت کلمات[۲]

تقریباً همه‌ی سیستم‌های موفق تشخیص گفتار[۳] از یادگیری ماشین به نحوی استفاده می‌کنند. برای مثال، سیستم Lee Sphinx) (1989 استراتژی تشخیص صداهای اولیه[۴] و کلمات را از سیگنال‌های مشاهده شده یاد می‌گیرد. متدهای شبکه‌های عصبی (Waibel 1989) و متدهای یادگیری مدل‌های پنهان مارکوف (Lee 1989) برای تغییر سیستم برای حساس بودن به افراد مختلف، فرهنگ لغات مختلف، میکروفون‌های مختلف، صدا با نویز و ... مؤثر است. متدهای مشابهی کاربرد مشابهی در بسیاری از سیستم‌های تفسیر سیگنال[۵] دارند.

- یادگیری هدایت یک اتومبیل.

متدهای یادگیری ماشین در آموزش اتومبیل‌های خودکار[۶] در انواع جاده‌ها و خیابان‌ها به درستی به کار رفته‌اند. برای مثال، سیستم ALVINN (Pomereau 1989) برای هدایت اتومبیل در سرعت ۷۰ مایل بر ساعت و طول ۹۰ مایل در میان اتومبیل‌های دیگر به درستی عمل کرده است. تکنیک‌های مشابهی کاربردهای احتمالی در بسیاری از مسائل حسگری[۷] دارند.

---

[۱] recogonize handwritten words
[۲] spoken words
[۳] speech recognition
[۴] phonemes
[۵] signal-interpretation
[۶] computer-controlled
[۷] sensor-based

- یادگیری دسته‌بندی ساختارهای نجومی جدید.

یادگیری ماشین در پایگاه داده‌های بزرگ مختلفی برای یادگیری نظم‌های کلی به کار رفته‌اند. برای مثال، الگوریتم‌های درخت یادگیری در ناسا[۱] برای یادگیری چگونگی دسته‌بندی اشیاء اسمانی در تحقیق (second Palomar Observatory Sky Survey 1995) به کار رفته‌اند.

- یادگیری بازی‌های کلماتی نظیر تخته‌نرد

موفق‌ترین برنامه‌های بازی‌هایی مثل تخته‌نرد بر پایه‌ی الگوریتم‌های یادگیری ماشین نوشته شده‌اند. برای مثال، بهترین برنامه‌ی جهان برای تخته‌نرد، TD-Gammon (Tesauro 1992, 1995) استراتژی بازی را با یک‌میلیون بازی کردن در مقابل خودش یاد می‌گیرد. این برنامه هم اکنون در مسابقات جهانی با انسان‌ها مسابقه می‌دهد. تکنیک‌های مشابه در بسیاری از مسائل کاربردی که در آن‌ها فضای جستجو بسیار بزرگ است را می‌توان به کار برد.

جدول ۱,۱ چندین نمونه کاربرد موفق یادگیری ماشین.

- هوش مصنوعی

یادگیری نمایش نمادی مفاهیم. یادگیری ماشین به نگاه جستجو. یادگیری به عنوان روشی برای بهبود حل مسئله. استفاده‌ی همزمان از دانش قبلی و داده‌های آموزشی برای یادگیری.

- متدهای بیزی

قضیه‌ی بیز به عنوان پایه‌ی محاسبه‌ی احتمالات فرضیه‌ها. دسته‌بندی کننده‌ی ساده‌ی بیز. الگوریتم‌های تخمین مقدار متغیرهای نامعلوم.

- تئوری پیچیدگی محاسباتی[۲]

محدودیت‌های تئوری موجود بر روی پیچیدگی مسائل یادگیری مختلف، که در غالب پیچیدگی محاسباتی، تعداد نمونه‌های آموزشی، تعداد خطای قابل‌تحمل و ... بیان می‌شود.

- تئوری کنترل (پیش‌بینی)[۳]

رویه‌هایی[۴] که یاد می‌گیرد تا مقادیر از پیش تعیین‌شده‌ای را بهینه و مرحله‌ی بعدی فرآیند که کنترل می‌شود را پیش‌بینی کند.

- تئوری اطلاعات[۵]

معیار آنتروپی و مفهوم اطلاعات. روش کوتاه‌ترین توضیح در یادگیری. کد سازی بهینه و رابطه‌ی آن با سری آموزشی بهینه برای توصیف یک فرضیه.

- فلسفه

تیغ occam، که توصیه می‌کند بهترین فرضیه ساده‌ترین آن‌هاست. بررسی توجیه برای تعمیم فرای داده‌های آموزشی مشاهده شده.

- روانشناسی و عصب‌شناسی

---

۱ NASA
۲ computational complexity theory
۳ control theory
۴ Procedures
۵ information theory

قانون قدرت تمرین[۱] ، که می‌گوید که سرعت عکس‌العمل انسان بر اثر تمرین بر روی مسائل مختلف یادگیری بهبود می‌یابد. تحقیقات عصب‌شناسی پایه‌ی مدل‌های شبکه‌های عصبی مصنوعی در یادگیری را تشکیل می‌دهند.	
• آمار	
توصیف ویژگی‌های خطا (مثل، بایاس و واریانس) که موقع تخمین دقت یک فرضیه بر اساس نمونه داده‌های محدود انجام می‌گیرد. بازه‌های اطمینان، آزمون‌های آماری.	

جدول ۱,۲ بعضی رشته‌های علمی و نمونه‌ای از تأثیرشان در یادگیری ماشین.

تعریف ما از یادگیری به اندازه‌ی کافی کلی است تا تمامی کارهایی که به طور کلی "یادگیری" نامیده می‌شود را در بر بگیرد. این تعریف به اندازه‌ی کافی نیز کلی هست تا برنامه‌های کامپیوتری‌ای که کارایی‌شان با تجربه بیشتر می‌شود را در بر بگیرد. برای مثال، یک پایگاه داده که به کاربرانش اجازه می‌دهد تا داده‌ها را تغییر دهند نیز با این تعریف ما از سیستم یادگیر تطابق دارد: زیرا که کارایی آن نیز با تجربه‌ای که حاصل تغییر داده‌های پایگاه داده افزایش می‌یابد. بدون نگرانی در شمول بیش از حد این تعریف می‌توان برنامه‌های یادگیر را برنامه‌هایی دانست که بر اثر تجربیات پیشرفت می‌کنند. در اینجا هدف از بحث بررسی مفهوم کلمه‌ی "یادگیری" نیست بلکه هدف در اینجا تعریف دقیق دسته‌ای از مسائل است که به نحوی به یادگیری مربوط می‌شوند. برای بررسی الگوریتم‌های حل چنین مسائلی و درک بهتر مبانی ساختاری مسائل و فرایندهای یادگیری به چنین تعریف دقیقی نیاز داریم.

۱,۲ طراحی یک سیستم یادگیری

برای به تصویر کشیدن بعضی از مشکلات طراحی و روش‌های یادگیری ماشین بیایید طراحی برنامه‌ای برای یادگیری بازی چکرز با هدف بازی در مسابقات چکرز را بررسی کنیم. واضح است که کارایی را درصد بازی‌های برده در این مسابقات تعیین می‌کنیم.

۱,۲,۱ انتخاب تجربیات آموزشی

اولین انتخاب طراحی، انتخاب نوع تجربیات آموزشی است که انتظار می‌رود سیستم با آن‌ها یاد بگیرد است. انتخاب نوع تجربیات آموزشی می‌تواند اثر چشم‌گیری در موفقیت یا شکست یادگیر داشته باشد. یکی از ویژگی‌های مهم تجربیات آموزشی مستقیم یا غیرمستقیم بودن آن است. برای مثال، در یادگیری بازی چکرز، ممکن است تجربیات آموزشی چینش‌های صفحه‌ی چکرز با حرکت مناسب مربوطه باشند، که نمونه‌ای از تجربیات آموزشی مستقیم است. اما ممکن است اطلاعات به طور غیرمستقیم باشد، مثلاً سری‌ای از حرکات یادگیر و نتیجه‌ی بازی باشد. در این حالت، درستی هر حرکت خاص در این بازی باید به طور غیرمستقیم از این حقیقت که نتیجه‌ی بازی برد یا باخت بوده استنباط شود. پس یادگیر با مسئله‌ی دیگری، ارزش‌دهی[۲] یا تعیین میزان تأثیر حرکات در نتیجه بازی مواجه است. نسبت دادن ارزش به حرکات نیز می‌تواند بسیار سخت باشد، زیرا که ممکن است حرکات ابتدایی بازی بسیار عالی بوده و اما نتیجه‌ی بازی باخت شده است. پس در حالت کلی یادگیری از تجربیات آموزشی مستقیم بسیار ساده‌تر از تجربیات غیرمستقیم است.

ویژگی مهم دوم تجربیات آموزشی درجه اختیار یادگیر در کنترل سری‌های نمونه‌های آموزشی است. برای مثال، ممکن است نمونه‌های ارائه شده به یادگیر توسط معلمی تعیین شود، یعنی معلمی چینش‌های صفحه را انتخاب کرده و آن‌ها را با حرکت متناسبشان به یادگیر بدهد. یا از

[۱] power low of practice

[۲] Credit assignment

طرف دیگر، یادگیر صفحاتی را که برایش ابهام دارند به معلم بدهد تا وی حرکت متناسبش را تعیین کند. یا حتی ممکن است یادگیر کنترلی هم بر چینش صفحات و هم به طور غیرمستقیم دسته‌بندی صفحات داشته باشد، برای مثال معلمی وجود نداشته باشد و برنامه در مقابل خودش بازی چکرز را انجام دهد. توجه دارید که در این صورت، یادگیر ممکن است انتخاب کند که وضعیت‌های جدیدی را که هنوز با آن مواجه نشده را بررسی کند یا در مقابل است وضعیت‌های گذشته‌اش را امتحان کند تا میزان امیدوار کننده بودن هر یک از وضعیت‌ها را معلوم کند. در فصول آتی تعدادی از تعریف مسئله‌های یادگیری شامل مسائلی که در آن نمونه‌های آموزشی به طور تصادفی و خارج از کنترل یادگیر انتخاب می‌شوند، مسائلی که یادگیر انواع مختلفی از آزمایش را به معلمی حرفه‌ای ارائه می‌کند و جواب را جویا می‌شود، و مسائلی که در آن یادگیر نمونه‌های آموزشی را با حرکت خودکار در محیط اطراف خود به دست می‌آورد را بررسی خواهیم کرد.

ویژگی مهم سوم تجربیات آموزشی، میزان نمایندگی آن از توزیع نمونه‌هایی است که برای تعیین کردن کارایی P سیستم نهایی استفاده می‌شود. در کل، زمانی که نمونه‌های آموزشی توزیعی مشابه نمونه‌های تست دارند یادگیری قابل‌اعتمادتر است. در مسئله‌ی یادگیری بازی چکرز ما، معیار کارایی P درصد بازی‌های برده در مسابقات جهانی است. اگر تجربیات آموزشی E فقط از بازی مقابل خود سیستم به دست آمده باشد، این خطر به وضوح موجود است که تجربیات آموزشی ممکن است نمونه کاملی از توزیع حالات ممکنی که بعداً در مسابقات سیستم با آن تست می‌شود نباشد. برای مثال، یادگیر ممکن است هیچ گاه با حالات بسیار وخیمی که بسیار در بازی با انسان به وجود می‌آید مواجه نشده باشد. در عمل، گاهی لازم است که یادگیری بر روی مجموعه‌ای از نمونه‌هایی که با نمونه‌های تست نهایی متفاوت‌اند آموزش داده شود (برای مثال ممکن است که مسابقات جهانی علاقه‌ای به آموزش سیستم ما نداشته باشد). در چنین شرایطی مشکل‌زا هستند زیرا که تسلط بر توزیعی از نمونه‌ها الزاماً به کارایی بالا در توزیع دیگر نمی‌انجامد. همان طور که خواهیم دید، مهم‌ترین تئوری یادگیری ماشین به این فرض اساسی وابسته است که توزیع نمونه‌های آموزشی مشابه توزیع نمونه‌های تست است. بر خلاف این فرض که برای رسیدن به نتایج تئوری انجام می‌دهیم، باید در نظر داشت که گاهی در عمل این فرض کاملاً برقرار نیست.

برای ادامه‌ی طراحی بیایید فرض کنیم که سیستم از بازی مقابل خودش آموزش می‌بیند. این فرض از این جهت که الزام وجود معلم خارجی را از بین می‌برد مزیت دارد، از طرف دیگر سیستم می‌تواند تا جایی که زمان اجازه می‌دهد داده‌ی آموزشی ایجاد کند. حال مسئله به طور کامل تعریف شده است:

مسئله‌ی یادگیری بازی چکرز:

- عمل T: بازی کردن چکرز
- کارایی P: درصد بازی‌های برده در مسابقات
- تجربیات آموزشی E: بازی‌هایی که در مقابل خودش انجام می‌دهد

حال برای کامل کردن طراحی سیستم یادگیری باید موارد زیر را معلوم کنیم،

۱. نوع دقیق دانشی که قصد داریم سیستم یاد بگیرد
۲. نمایشی برای این دانش هدف
۳. روشی برای یادگیری

۱,۲,۲ انتخاب تابع هدف

مرحله‌ی بعدی طراحی تعیین دقیق نوع دانشی و چگونگی استفاده از این دانش برای بهبود کارایی سیستم است. بیایید با یک برنامه‌ی بازی چکرز شروع کنیم که حرکات مجاز را در هر چینش صفحه تشخیص می‌دهد. حال کافی است فقط راهی برای تعیین بهترین حرکت در میان حرکات مجاز یاد بگیریم. این کار یادگیری نماینده‌ی دسته‌ی بزرگی از کارهای یادگیری است که در آن تعدادی عمل مجاز در دسترس است و فضای جستجو بسیار بزرگی نیز مشخص شده است اما روش پیدا کردن بهترین حرکت معلوم نیست. بسیاری از مسائل بهینه‌سازی[۱] از این دسته مسائل‌اند، مسائلی مثل برنامه‌ریزی و کنترل خط تولید که در آن‌ها مراحل تولید مشخص‌اند اما بهترین استراتژی ترتیب آن‌ها مشخص نیست مثالی از این گونه مسائل است.

با این تعریف مسئله، باید یاد بگیریم تا از میان حرکت‌های مجاز یکی را انتخاب کنیم، واضح‌ترین گزینه برای نوع اطلاعات یادگیری، یک برنامه یا تابع است که بهترین حرکت را با داشتن چینش صفحه پیدا می‌کند. بیایید این تابع را ChooseMove بنامیم و $ChooseMove: B \rightarrow M$. توجه دارید که این تابع چینشی مجاز از چینش‌های مجاز صفحه B را دریافت کرده و حرکتی را از میان حرکات مجاز M به عنوان خروجی می‌دهد. در سراسر بحث یادگیری ماشین، همیشه بد نیست که مسئله‌ی بهینه‌سازی کارایی P در عمل T را به مسئله‌ی یادگیری یک تابع مثل ChooseMove کاهش دهیم. بنابراین انتخاب تابع هدف یکی از انتخاب‌های کلیدی طراحی خواهد بود.

با وجود اینکه تابع ChooseMove در مثال ما بسیار ساده تعریف می‌شود اما یادگیری آن با داشتن تجربیات آموزشی غیرمستقیم برای سیستم بسیار سخت خواهد بود. می‌توان بجای چنین تابعی، تابعی دیگر، که در این تعریف مسئله یادگیری‌اش بسیار ساده‌تر است، را یاد گرفت، این تابع تابعی ارزیاب[۲] است که به هر چینش صفحه یک ارزش یا امتیاز نسبت می‌دهد. بیایید این تابع را V بنامیم و با توجه به نام‌گذاری‌های قبلی خواهیم داشت، $V: B \rightarrow \Re$، یعنی تابع V به هر چینش صفحه‌ی مجاز یک عدد حقیقی نسبت می‌دهد ($\Re$ برای نماد اعداد حقیقی به کار می‌رود). ما می‌خواهیم که تابع هدف V به چینش‌های بهتر صفحه عددی بیشتر نسبت دهد. اگر سیستم بتواند با موفقیت چنین تابع V ای را یاد بگیرد می‌تواند به راحتی بهترین حرکت در هر چینش صفحه را انتخاب کند. این کار را می‌توان با تولید چینش‌های آتی صفحه که پس از هر یک از حرکات مجاز ایجاد می‌شود و مقایسه‌ی مقادیر V آن‌ها انجام داد (حرکت نظیر بهترین چینش بهترین حرکت مجاز است).

اما دقیقاً چگونه می‌توان مقدار تابع هدف V را برای هر چینش صفحه مشخص کرد؟ البته، هر تابع ارزیابی‌ای که به چینش‌های بهتر عدد بیشتری نسبت دهد قابل‌قبول است. با این وجود بهتر است که تابع هدفی خاص را در میان تمامی توابعی که حرکت بهینه را تشخیص می‌دهند برای V مشخص کنیم. همان طور که بعداً نیز خواهیم دید، بهتر است الگوریتمی برای یادگیری طراحی شود. پس بیایید مقدار V(b) را که b چینشی از مجموعه چینش‌های ممکن B است را به صورت زیر تعریف کنیم:

۱. اگر b چینشی انتهایی برنده بود، V(b)=100
۲. اگر b چینشی انتهایی بازنده بود، V(b)=-100
۳. اگر b چینشی انتهایی مساوی بود، V(b)=0
۴. اگر b چینشی در انتهای بازی نبود، V(b)=V(b′) که b′ بهترین چینش صفحه‌ی ممکن حاصل از چینش b با بازی بهینه تا آخر بازی (با فرض اینکه حریف نیز بهینه بازی کند) خواهد بود.

---

[۱] optimation
[۲] Evaluation function

با اینکه تعریف بازگشتی از مقدار V(b) برای هر چینش صفحه‌ی b تعیین می‌کند، این تعریف برای بازیکن چکرز ما قابل استفاده نخواهد بود زیرا که مقادیر قابل محاسبه نیست. مگر در حالت‌های انتهایی (موارد ۱ تا ۳) که در آن‌ها بازی تمام شده است و مشخص کردن V(b) ارزشی ندارد، مشخص کردن مقدار V(b) برای یک چینش صفحه‌ی خاص (مورد ۴) به جستجو برای سری بهینه‌ای از حرکات می‌انجامد که بازی را به آخر می‌رساند! چون این تعریف برای برنامه‌ی چکرز ما قابل محاسبه نیست، این تعریف تعریفی غیرعملی[۳] نامیده می‌شود. هدف یادگیری در این مرحله پیدا کردن تعریفی عملی[۴] از V است؛ تعریفی که بتوان آن را در برنامه‌ی بازی چکرز برای ارزیابی چینش‌ها و انتخاب حرکات به کار گرفت.

بنابراین، کار یادگیری را در این مثال به مسئله‌ی پیدا کردن تعریفی عملی از تابع هدف V کاهش دادیم. یادگیری فرم دقیقی از V در حالت کلی خیلی سخت خواهد بود. در واقع، گاهی اوقات فقط انتظار داریم که الگوریتم‌های یادگیری تخمینی از تابع هدف را پیدا کنند و به همین دلیل فرایند یادگیری تابع هدف تخمین[۵] تابع هدف نیز نامیده می‌شود. در بحث فعلی از نماد $\hat{V}$ برای تابع یادگیری شده (تخمین تابع هدف V) استفاده می‌کنیم.

۱,۲,۳ انتخاب نحوه‌ی نمایش تابع هدف

حال که تابع هدف V را مشخص کردیم، باید نمایشی انتخاب کرده تا برنامه بتواند تابع $\hat{V}$ را با آن نشان دهد. مثل انتخاب‌های قبلی طراحی در اینجا نیز با گزینه‌های بسیاری مواجهیم. برای مثال، می‌توانیم به برنامه اجازه دهیم که $\hat{V}$ را با جدول بزرگی از مقادیر برای هر یک از چینش‌های صفحه نشان دهد. یا می‌توانیم به آن اجازه دهیم تا $\hat{V}$ را با مجموعه‌ای از قوانین که با ویژگی‌های چینش صفحه مطابقت دارد یا تابعی درجه دو از ویژگی‌هایی از پیش تعریف شده یا یک شبکه‌ی عصبی مصنوعی نمایش دهد. در کل، این انتخاب نمایش شامل یک مقایسه‌ی مهم است. در یک طرف، سعی می‌کنیم نمایشی که انتخاب کنیم کاملاً شامل باشد تا بتوان آن را به اندازه‌ی کافی به تعریف ایده آل V نزدیک کرد. از طرف دیگر، با شامل‌تر بودن این نمایش تعداد داده‌های آموزشی که برنامه نیاز خواهد داشت تا میان فرضیه‌ها بتواند مناسب‌ترین را انتخاب کند بیشتر خواهد شد. خلاصه اینکه، بیایید یک نمایش ساده را انتخاب کنیم: V را به عنوان ترکیب خطی[۶] ویژگی‌های در نظر می‌گیریم:

- x_1: تعداد مهره‌های سیاه در صفحه
- x_2: تعداد مهره‌های قرمز در صفحه
- x_3: تعداد مهره‌های شاه سیاه در صفحه
- x_4: تعداد مهره‌های شاه قرمز در صفحه
- x_5: تعداد مهره‌های سیاه تهدید شده توسط قرمز (که سیاه می‌تواند در حرکت بعدی آن را بگیرد)
- x_6: تعداد مهره‌های قرمز تهدید شده توسط سیاه

بنابراین برنامه تابع $\hat{V}(b)$ را با تابعی خطی به فرم زیر بیان خواهد کرد:

---

[۳] nonoperational definition
[۴] operational definition
[۵] approximation
[۶] linear combination

$$\hat{V}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

در این رابطه w_0 تا w_6 ضرایب عددی یا همان وزن‌ها هستند که توسط الگوریتم یادگیری تعیین می‌شوند. مقادیر w_1 تا w_6 اهمیت نسبی متغیرهای مختلف صفحه مشخص می‌کنند و w_0 نیز ثابتی به این مقدار صفحه اضافه می‌کند.

به طور خلاصه، با انتخاب‌های طراحی‌مان تا به حال، نوع تجربیات یادگیر، تابع هدف تخمینی و فرمی برای نمایش آن بوده است. مسئله در حال حاضر به شکل زیر است:

طراحی میانی برنامه‌ی یادگیری چکرز:

- کار T: بازی چکرز
- کارایی P: درصد بازی‌های برده در مسابقات
- تجربیات آموزشی E: بازی‌هایی که برنامه مقابل خود انجام می‌دهد
- تابع هدف: $V : Board \rightarrow \Re$
- نمایش تابع هدف:

$$\hat{V}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

سه قسمت اول کار یادگیری را مشخص می‌کنند در حالی که دو قسمت انتهایی مربوط به انتخاب‌های طراحی ما برای پیاده‌سازی برنامه‌ی یادگیری هستند. توجه کنید که اضافه کردن این قسمت‌ها فقط برای کاهش مسئله‌ی یادگیری استراتژی بازی چکرز به مسئله‌ی یادگیری مقادیر ضرایب w_0 تا w_6 موجود در نمایش تابع هدف است.

۱,۲,٤ انتخاب یک الگوریتم تخمین تابع

برای یادگیری تابع هدف $\hat{V}$ نیاز به مجموعه‌ای از نمونه‌های آموزشی داریم، که هر کدام یک چینش صفحه‌ی b و یک مقدار یادگیری $V_{train}(b)$ برای b است. به عبارت دیگر، هر نمونه‌ی آموزشی زوج مرتبی به شکل $< b, V_{train}(b) >$ است. برای مثال، نمونه‌ی زیر چینشی را نشان می‌دهد که سیاه بازی را برده است (توجه دارید که $x_2 = 0$ بدین معناست که قرمز مهره‌ی دیگری در صفحه ندارد) بنابراین مقدار V در این نمونه 100+ خواهد بود.

$$<< x_3 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0 >, +100 >$$

در زیر فرایندی را که ابتدا چنین نمونه‌های آموزشی‌ای را از تجربیات آموزشی غیرمستقیم استخراج می‌کنند و سپس وزن‌های w_i را برای نمونه‌های آموزشی پیدا می‌کنند را توضیح خواهیم داد.

۱,۲,٤,۱ تخمین مقادیر آموزشی

با توجه به فرضی که در مسئله‌ی یادگیری کردیم، تنها اطلاعات آموزشی موجود برای یادگیر این است که آیا بازی نتیجه‌ی بازی برد بوده یا باخت. در مقابل، نمونه‌های آموزشی‌ای لازم داریم که به هر یک از چینش‌های صفحه یک امتیاز نسبت دهند. با وجود اینکه نسبت دادن مقدار به چینش‌هایی که انتهای بازی هستند بسیار ساده است اما نسبت دادن مقادیر آموزشی عددی به چینش‌هایی که در وسط بازی قرار دارند، اصلاً ساده نیست. البته این حقیقت که نتیجه‌ی بازی برد یا باخت بوده نشان نمی‌دهد که تک‌تک چینش‌های صفحه بازی خوب یا بد بوده است. برای

مثال اگر برنامه بازی را ببازد، ممکن است بدین معنا باشد که چینش‌های ابتدایی صفحه بازی باید عدد بیشتری داشته و چینش‌های انتهایی عدد کمتری دارد و باخت نتیجه‌ی سری حرکات ضعیف میانی‌ای بوده است.

برخلاف ابهام ذاتی در تخمین مقادیر آموزشی چینش‌های میانی بازی، یک روش ساده بسیار مفید عمل می‌کند. این روش مقدار آموزشی $V_{train}(b)$ را برای هر چینش میانی b، $\hat{V}(successor(b))$ مقداردهی می‌کند، در این مقدار $\hat{V}$ تخمین فعلی یاد گرفته شده از V و successor(b) نیز چینشی است که بعد از حرکت برنامه دوباره نوبت به وی می‌رسد (چینشی که پس از حرکت برنامه و حرکت حریف ایجاد می‌شود). این قانون برای تخمین مقادیر آموزشی را می‌توان به صورت زیر بیان کرد:

قانون تخمین مقادیر آموزشی:

$$V_{train}(b) \leftarrow \hat{V}\big(successor(b)\big) \qquad (1.1)$$

با وجود اینکه استفاده از $\hat{V}$ تخمینی (که خود از همین داده‌ها تخمین زده می‌شود) برای تخمین جدید مقادیر عجیب به نظر می‌رسد، اما این روش طبق تجربه موجه است. توجه دارید که از مقدار successor(b) برای تخمین مقدار چینش b استفاده می‌کنیم. شهوداً واضح است که دقت $\hat{V}$ در نزدیکی چینش‌های انتهایی افزایش می‌یابد. در واقع در شرایطی (که در فصل ۱۳ بحث خواهد شد) روش تخمین تکراری مقادیر آموزشی بر اساس تخمین چینش‌های successor ثابت می‌شود که به V_{train} میل خواهد کرد.

۱,۲,٤,۲ تنظیم وزن‌ها

تنها کار باقی مانده معین کردن الگوریتم یادگیری برای انتخاب وزن‌های w_i به صورتی است که بهترین تناسب را با نمونه‌های آموزشی $\{< b, V_{trian}(b) >\}$ داشته باشد است. به عنوان اولین مرحله، ابتدا باید تعریف کنیم که منظور از بهترین تناسب با داده‌های آموزشی چیست. یکی از روش‌های ممکن تعریف این بهترین فرضیه، یا بهترین مجموعه وزن‌ها به صورتی است که خطای مربعی E بین مقادیر آموزشی و مقادیر تخمینی $\hat{V}$ را مینیمم کنیم.

$$E \equiv \sum_{<b,V_{train}(b)> \in training\ examples} \Big(V_{train}(b) - \hat{V}(b)\Big)^2$$

بنابراین ما به دنبال وزن‌هایی هستیم، یا به طور مشابه به دنبال $\hat{V}$ ی هستیم که مقدار E را برای نمونه‌های آموزشی مشاهده شده مینیمم کند. در فصل ۶ ثابت می‌کنیم که در مسئله‌هایی مثل مسئله‌ی ما آن مینیمم کردن مجموع خطای مربعی متناظر با پیدا کردن محتمل‌ترین فرضیه با داشتن داده‌های آموزشی است.

الگوریتم‌های بسیاری برای پیدا کردن وزن‌های توابع خطی که E را مینیمم می‌کنند وجود دارد. در این حالت الگوریتمی مورد نیاز است که مرحله به مرحله با افزایش نمونه‌های آموزشی در وزن‌ها تجدید نظر کند و همچنین نسبت به خطای تخمین مقادیر نمونه‌های آموزشی حساسیت کمی داشته باشد. یکی از این الگوریتم‌ها، الگوریتم کمترین خطای مربعی یا LMS[۷] نامیده می‌شود. این الگوریتم برای هر نمونه‌ی آموزشی مشاهده شده وزن‌ها را به اندازه‌ی کوچک در جهتی که خطا را برای نمونه‌ی آموزشی کم می‌کند تغییر خواهد داد. همان طور که در فصل ۴ نیز

[۷] Least Mean Squars (LMS)

بررسی خواهیم کرد، این الگوریتم را می‌توان جستجوی شیب نزول تصادفی‌ای در فضای فرضیه‌های ممکن (مقادیر مختلف وزن‌ها) برای مینیمم کردن خطای مربعی E دانست . الگوریتم LMS به فرم زیر تعریف می‌شود:

قانون تغییر وزن LMS.

برای هر نمونه‌ی آموزشی $< b, V_{train}(b) >$

- از وزن‌های فعلی برای محاسبه‌ی $\hat{V}(b)$ استفاده کن.
- برای هر وزن w_i، تغییر زیر را اعمال کن

$$w_i \leftarrow w_i + \eta \left(V_{train}(b) - \hat{V}(b) \right) x_i$$

در اینجا η ثابت کوچکی (مثل ۰,۱) است که اندازه‌ی تغییر وزن را متعادل می‌کند. برای درک شهودی اینکه چرا این قانون تغییر وزن درست کار می‌کند، توجه کنید که زمانی که خطای $\left(V_{train}(b) - \hat{V}(b) \right)$ صفر است، وزن‌ها تغییری نخواهند کرد و زمانی که $\left(V_{train}(b) - \hat{V}(b) \right)$ مثبت است (برای مثال $\hat{V}(b)$ کمتر از انتظار است)، به هر وزن به نسبتی افزایش خواهد داد. این عمل مقدار $\hat{V}(b)$ را افزایش داده و در نهایت میزان خطا کمتر می‌شود. توجه داشته باشید که اگر مقدار ویژگی x_i صفر باشد، مستقل از این که خطا چه مقدار باشد وزن تغییری نخواهد کرد، بنابراین، تنها وزن‌ها نظیر متغیرهایی تغییر خواهند کرد که واقعاً در صفحه‌ی بازی اتفاق می‌افتند. جالب است که، اثبات می‌شود که این متد تنظیم وزن ساده حتماً به کمترین خطای مربعی تقریبی برای مقادیر V_{train} میل خواهد کرد (فصل ۴).

۱,۲,۵ طراحی نهایی

طراحی نهایی سیستم یادگیری چکرز را می‌توان با ۴ قسمت[۸] برنامه نشان داد که پایه‌ی اصلی بسیاری از سیستم‌های یادگیری هستند. این چهار قسمت در شکل ۱,۱ به طور خلاصه نشان داده شده‌اند:

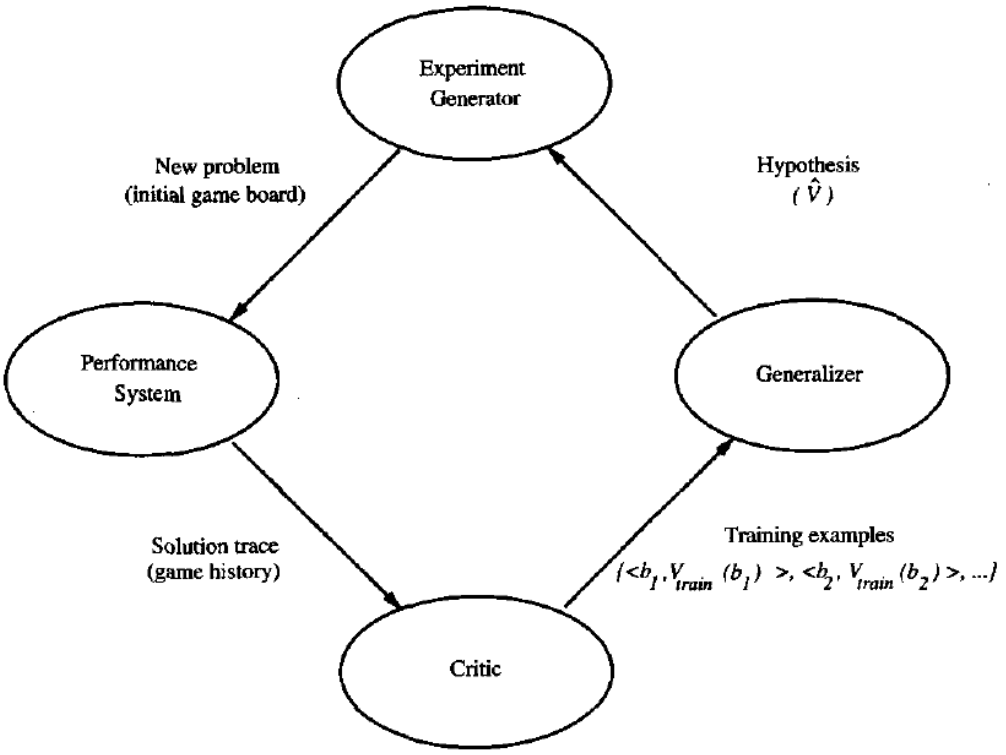
- **سیستم کارایی** قسمت است که باید مسئله‌ی پیدا کردن کارایی را حل کند، در مثال چکرز، این کار باید با استفاده از تابع هدف‌های یاد گرفته شده انجام شود. این قسمت نمونه‌ای از مسئله‌ای جدید (بازی جدید) به عنوان ورودی دریافت کرده و مسیری[۹] برای حل آن [۱۰] خروجی می‌دهد. در مثال ما، استراتژی سیستم کارایی در انتخاب حرکت بعدی در هر مرحله توسط تابع $\hat{V}$ مشخص می‌شود. بنابراین انتظار داریم که کارایی سیستم با افزایش دقت این تابع ارزیابی افزایش یابد.
- **کارشناس** مسیری از حرکات بازی را دریافت کرده و آن‌ها را به مجموعه‌ای از نمونه‌های آموزشی تبدیل می‌کند و خروجی می‌دهد. همان طور که در شکل نیز نشان داده شده است، هر نمونه‌ی آموزشی در این مثال متناسب با چینشی از صفحه در مسیر بازی و مقادیر تخمینی V_{train} شان است. در مثال ما، همان قانون یادگیری رابطه‌ی ۱,۱ است.
- **تامیم دهنده** مجموعه‌ای از نمونه‌های آموزشی را دریافت کرده و فرضیه‌ای متناسب با آن خروجی می‌دهد، این فرضیه همان تخمین تابع هدف است. این قسمت نمونه‌های آموزشی محدود را تامیم می‌دهد، و فرضیه‌ای که تابعی کلی است و این مجموعه و

---

[۸] module
[۹] trace
[۱۰] game history

دیگر نمونه‌ها را می‌پوشاند ارائه می‌کند. در مثال ما، تامیم دهنده الگوریتم LMS بود و خروجی آن نیز $\hat{V}$ بود که با وزن‌های w_0 تا w_6 مشخص می‌شد.

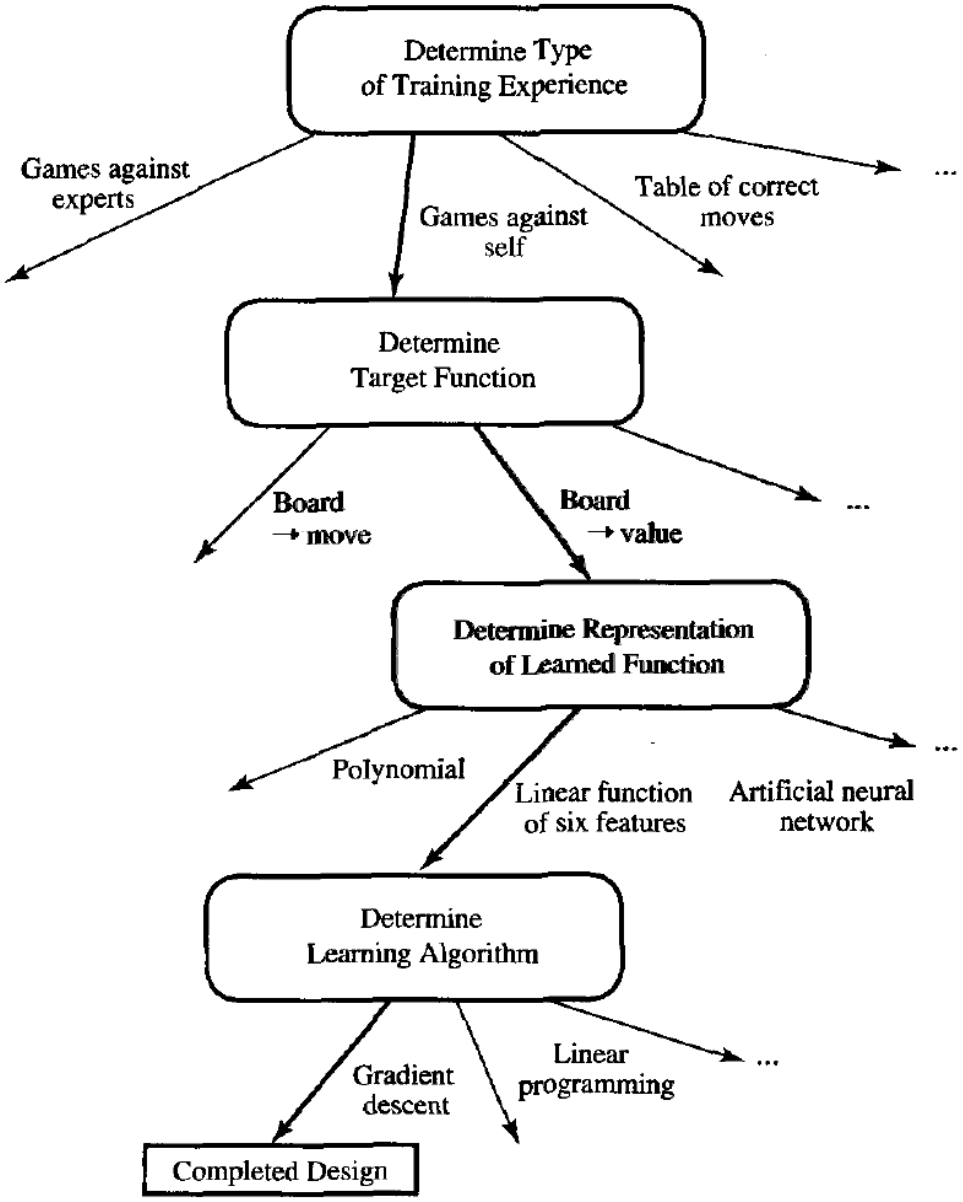
- **ایجاد کننده‌ی تجربه** فرضیه‌ی فعلی (تابعی که تا کنون یاد گرفته‌ایم) را به عنوان ورودی دریافت می‌کند و مسئله‌ای جدید ایجاد می‌کند (صفحه‌ای جدید ایجاد می‌کند) تا سیستم کارایی در آن به کاوش بپردازد. نقش این سیستم انتخاب مسئله‌های تمرینی جدیدی که سرعت یادگیری را به حداکثر برساند. در مثال ما، ایجاد کننده‌ی تجربه یک استراتژی بسیار ساده را دنبال می‌کرد: همیشه یک چینش صفحه‌ی ثابت را برای ایجاد بازی جدید انتخاب می‌کرد. در استراتژی‌های پیچیده‌تر را می‌توان برای کاوش ناحیه‌های خاص فضای چینش صفحه به کار برد.



شکل ۱,۱ طراحی نهایی برنامه یادگیری چکرز.

انتخاب‌ها طراحی‌ای که برای طراحی برنامه‌ی بازی چکرز انجام دادیم ویژگی‌های دقیق سیستم‌های کارایی، کارشناس، تامیم دهنده و ایجاد کننده‌ی تجربه، را تعیین می‌کند. بسیاری از سیستم‌های یادگیری ماشین را می‌توان در فرم همین چهار قسمت بیان کرد.

ترتیب انتخاب گزینه‌های طراحی برای بازی چکرز در شکل ۱,۲ به طور خلاصه آورده شده است. این انتخاب‌های طراحی کار یادگیری را از چندین نظر محدود کرده است. برای مثال، نوع دانشی که ذخیره می‌شود را به تابع خطی محدود کرده‌ایم. علاوه بر این، تابع خطی را فقط تابع ۶ متغیر خاص از صفحه‌ی بازی فرض کرده‌ایم. اگر تابع هدف V را بتوان با ترکیبی خطی از ویژگی‌ها نمایش داد، برنامه‌ی ما با احتمال خوبی V را تخمین خواهد زد. اما اگر V را نتوان با ترکیب خطی این متغیرها نشان داد در بهترین حالت می‌توان از برنامه انتظار داشت که تقریب خوبی از آن را یاد بگیرد. زیرا که هیچ برنامه‌ای نمی‌تواند چیزی را که نمی‌تواند نمایش دهد یاد بگیرد.



شکل ۱,۲ خلاصه‌ی انتخاب‌های طراحی برنامه‌ی بازی چکرز.

بیایید فرض کنیم که تقریب خوبی از V را بتوان با این فرم نمایش داد. حال این سؤال مطرح خواهد بود که آیا این تکنیک‌های یادگیری تضمین می‌کنند که در صورت وجود این تقریب آن را پیدا کنند. فصل ۱۳ بررسی ی تئوری انجام می‌دهد که برای تحت شرایط محدود کننده‌ای، روشی مشابه این روش واقعاً به سمت تابع ارزیابی میل خواهد کرد. خوشبختانه در نتایج عملی مشاهده می‌شود که حتی هنگامی که از محدوده‌ی شرایط اثبات خارج می‌شویم معمولاً این روش برای یادگیری تابع ارزیابی موفقیت‌آمیز است.

آیا برنامه‌ای که ما طراحی می‌کنیم به اندازه‌ی کافی قوی خواهد بود تا بتواند بازیکنی جهانی را ببرد؟ احتمالاً خیر. این به خاطر این است که نمایش خطی تابع $\hat{V}$ بسیار ساده است و نمی‌تواند جزئیات بازی را تعیین کند. با این وجود، با در نظر گرفتن نمایشی پیچیده‌تر برای تابع هدف، این روش کلی می‌تواند بسیار موفقیت‌آمیز باشد. برای مثال (Tesauro 1992, 1995) طرحی برای برنامه‌ای که یاد می‌گیرد تخته‌نرد بازی کند را با یادگیری تابع ارزیابی مشابهی بر روی وضعیت‌ها را ارائه می‌کند. برنامه‌ی وی تخمین تابع یاد گرفته شده را با استفاده از شبکه‌ای

عصبی که ویژگی‌های کامل وضعیت صفحه را به جای زیرمجموعه‌ای از ویژگی‌های صفحه دریافت می‌کند نمایش می‌دهد. بعد از آموزش بر روی یک‌میلیون بازی آموزشی خودساخته[۱۱] برنامه‌ی وی توانست در مقابل بازیکنان سطح بالای تخته‌نرد بازی کند.

البته می‌توانستیم الگوریتم‌های دیگری را برای کار یادگیری بازی چکرز طراحی کنیم. برای مثال، ممکن بود نمونه‌های آموزشی را ذخیره کرده، و در وضعیت‌های جدید در درون مجموعه‌ی ذخیره شده به دنبال نمونه‌های مشابه بگردیم (الگوریتم nearest-neighbor، فصل ۸). یا می‌توانستیم تعداد زیاد از برنامه‌های بازی چکرز را ایجاد کرده و اجازه‌ی بازی به آن‌ها بدهیم و موفق‌ترین آن‌ها را حفظ کرده و با این مجموعه با جهش[۱۲] و ترکیب برنامه‌ها را تکامل دهیم (الگوریتم‌های ژنتیک، فصل ۹). به نظر می‌رسد انسان‌ها روشی متفاوت برای یادگیری استفاده می‌کنند، در این روش آن‌ها شرایط را بررسی کرده و توضیحات و دلایلی برای موفقیت یا شکست بازی ایجاد می‌کنند (یادگیری توضیحی، فصل ۱۱). طراحی ما یکی از طراحی‌های ممکن است و برای آشنایی با بحث و انتخاب‌های طراحی متد یادگیری دسته‌ای از مسائل آورده شده است.

۱,۳ دورنما و مشکلات یادگیری ماشین

یکی از نگاه‌های یادگیری ماشین، جستجویی میان فضای فرضیه‌ای نسبتاً بزرگ فرضیه‌های ممکن برای مشخص کردن بهترین فرضیه با توجه به داده‌های آموزشی موجود و دانش قبلی است. برای مثال، فضای فرضیه‌ای تمامی فرضیه‌هایی خروجی یادگیر بازی چکرز (که در بالا طراحی کردیم) را در نظر بگیرید. این فضای فرضیه‌ای شامل تمامی توابع ارزیابی‌ای می‌شود که می‌توان آن‌ها را در قالب w_0 تا w_6 بیان کرد. پس کار یادگیر جستجو در میان این فضای فرضیه‌ای وسیع برای پیدا کردن سازگارترین فرضیه با نمونه‌های آموزشی موجود است. الگوریتم LMS با تکرار تغییر وزن‌ها و تصحیح تخمین‌های اشتباه تابع در هر مرحله به این تابع ارزیابی دست پیدا می‌کند. این الگوریتم را می‌توان هنگامی که نمایش فرضیه یادگیر با پارامترهای پیوسته است کار برد.

بسیاری از فصول این کتاب الگوریتم‌هایی را ارائه می‌کنند که فضای فرضیه‌ای تعریف شده با استفاده از نمایشی خاص (مثل توابع خطی، توصیف منطقی، درخت تصمیم و شبکه‌های عصبی) را جستجو می‌کنند. این نمایش‌های متفاوت فرضیه‌ها برای یادگیری انواع مختلف توابع هدف است. در هر یک از این نمایش فرضیه‌ها، الگوریتم یادگیری مناسب از ساختاری نمایش برای ترتیب جستجو در فضای فرضیه‌ای کمک می‌گیرد.

در تمام طول این کتاب، از این نگاه به مسائل یادگیری برای دسته‌بندی متدهای یادگیری بر اساس استراتژی‌های جستجو و ساختار فضای فرضیه‌ای مورد جستجو کمک می‌گیریم. همچنین این نگاه را برای بررسی رسمی روابط بین اندازه‌ی فضای فرضیه‌ای مورد جستجو، تعداد نمونه‌های آموزشی موجود و اطمینان تعمیم فرضیه نهایی بر روی‌داده‌های جدید کار می‌بریم.

۱,۳,۱ مشکلات یادگیری ماشین

مسئله‌ی مطرح شده چکرز سؤالات کلی‌ای درباره‌ی یادگیری ماشین ایجاد می‌کند. یادگیری ماشین و اکثر متن این کتاب برای جواب به چنین سؤالاتی است:

---

[۱۱] self-generated

[۱۲] mutate

- چه الگوریتم‌هایی برای یادگیری کلی توابع هدف از نمونه‌های آموزشی خاص وجود دارد؟ در چه شرایطی یک الگوریتم خاص با داشتن نمونه‌های آموزشی کافی به تابع مورد نظر میل می‌کند؟ چه الگوریتم‌هایی برای چه نوع از مسائل و نمایش‌ها کارایی بهتری دارند؟
- چه میزان داده‌ی آموزشی کافی است؟ چه محدودیت‌های کلی‌ای را می‌توان برای رابطه‌ی اطمینان فرضیه‌ها، میزان تجربیات آموزشی و ویژگی‌های فضای فرضیه‌ای یادگیر به دست آورد؟
- در چه شرایطی و چگونه دانش قبلی یادگیر می‌تواند به فرآیند یادگیری کمک کند؟ آیا دانش قبلی زمانی که کاملاً درست نیست نیز می‌تواند به فرآیند یادگیری کمک کند؟
- بهترین روش انتخاب تجربه‌ی آموزشی بعدی چیست، و چگونه این انتخاب این روش پیچیدگی یادگیری مسئله را تغییر می‌دهد؟
- بهترین راه کاهش کار یادگیری به یک یا چند مسئله‌ی تابع تخمین چیست؟ به عبارت دیگر، چه توابع خاصی را باید هدف یادگیری قرار داد؟ آیا می‌توان خود این فرایند را خودکار[۱۳] کرد؟
- چگونه یادگیر می‌تواند به طور خودکار نحوه‌ی نمایش را برای بهتر کردن قدرت نمایش و یادگیری تابع هدف تغییر دهد؟

۱,٤ این کتاب را چگونه بخوانیم

این کتاب شامل مقدمه‌ای بر الگوریتم‌ها و روش‌های ابتدایی یادگیری ماشین و نتیجه‌های تئوری از امکان[۱۴] یادگیری کارهای مختلف و ظرفیت‌های الگوریتم‌های خاص و نمونه‌های کاربردی یادگیری ماشین در جهان واقعی است. فصل‌های این کتاب را می‌توان به هر ترتیب دلخواه خواند، با این وجود بعضی وابستگی‌ها بین فصول اجتناب ناپذیرند. اگر این کتاب را برای سیلابس درسی استفاده می‌کنید، واقعاً توصیه می‌شود که ابتدا به فصول ۱ و ۲ پرداخته شود. به جز این دو فصل بقیه فصول را می‌توان تقریباً به هر ترتیب دلخواه ممکن خواند. برای کلاسی که یک ترم خواهد بود متن ۷ فصل کافی خواهد بود، البته فصول اضافی می‌تواند برای مطالعه‌ی آزاد گذاشته شود (که از اهمیت قابل توجهی برخوردارند). در زیر خلاصه‌ای از آنچه در هر فصل آورده شده آمده:

- فصل ۲ به یادگیری مفهوم بر پایه‌ی نمایش نمادین[۱۵] و نمایش منطقی است. همچنین در این فصل ترتیب کلی به جزئی فرضیه‌ها و بایاس استقرایی و اهمیتش بررسی شده است.
- فصل ۳ یادگیری درختی و مسئله‌ی overfit را بررسی می‌کند. همچنین تیغ occam قانونی که فرضیه‌های کوتاه‌تر را ترجیح می‌دهد– نیز آورده شده است.
- فصل ۴ به یادگیری شبکه‌های عصبی و الگوریتم Backpropagation و روش کلی شیب نزول می‌پردازد. این بخش شامل مثالی از پردازش اطلاعات تصویر (صورت انسان) نیز می‌شود. آدرس منابع داده‌ها و الگوریتم‌های اضافه نیز آورده شده است.
- فصل ۵ به مفاهیم پایه‌ای تئوری آمار و تخمین با تمرکز بر ارزیابی دقت فرضیه‌ها با استفاده از داده‌های محدود می‌پردازد. این فصل به بازه‌های اطمینان[۱۶] برای تخمین دقت فرضیه‌ها و متدهای مقایسه‌ی دقت متدهای مختلف یادگیری می‌پردازد.

---

[۱۳] automatic
[۱۴] feasibility
[۱۵] symbolic representation
[۱۶] confidence interval

- فصل ۶ به یادگیری بیزی در یادگیری ماشین می‌پردازد، این فصل به کاربرد بررسی بیزی هم برای بررسی الگوریتم‌های یادگیری ی غیر بیزی و هم برای الگوریتم‌های یادگیری بیزی، که احتمال فرضیه‌های را محاسبه می‌کنند، می‌پردازد. این فصل همچنین شامل مثالی از به کار بردن دسته‌بندی کننده‌ی ساده‌ی بیز در مسئله‌ی دسته‌بندی متون، با استفاده از برنامه و داده‌های اینترنت می‌شود.
- فصل ۷ نظریه‌ی یادگیری محاسباتی را پوشش می‌دهد. در این فصل به مدل یادگیری تقریباً درست (probably approximately correct (PAC)) و مدل یادگیری مرز خطا[۱۷] خواهیم پرداخت. علاوه بر این در این فصل به الگوریتم رأی‌گیری وزن‌دار[۱۸] نیز خواهیم پرداخت که روشی برای ترکیب الگوریتم‌های یادگیری است.
- فصل ۸ به متدهای یادگیری مبتنی بر نمونه‌ها می‌پردازد، این متدها شامل یادگیری نزدیک‌ترین همسایه‌ها، برازش وزن‌دار محلی و case-based reasoning می‌شود.
- فصل ۹ الگوریتم‌های یادگیری که با الهام از تکامل زیستی ساخته شده‌اند را بررسی می‌کند. این الگوریتم‌ها شامل الگوریتم‌های ژنتیک و برنامه‌نویسی ژنتیک می‌شوند.
- فصل ۱۰ الگوریتم‌هایی که برای یادگیری دسته قوانین، شامل روش‌های برنامه‌نویسی منطقی استقرایی برای horn clause های درجه اول، را پوشش می‌دهد.
- فصل ۱۱ به یادگیری توضیحی، متدی یادگیری که از دانش قبلی برای توضیح نمونه‌های آموزشی استفاده می‌کند و با این توضیحات بر روی نمونه‌های آموزشی تعمیم می‌دهد می‌پردازد.
- فصل ۱۲ روش‌های بهبود دقت فرضیه یادگیری با ترکیب دانش قبلی و نمونه‌های آموزشی را بحث خواهد کرد. در این فصل هم از الگوریتم‌های نمادی و هم از شبکه‌های عصبی استفاده خواهد شد.
- فصل ۱۳ به یادگیری تقویتی می‌پردازد، روشی که سیستم باید کارایی خود را طبق پاداش‌هایی که دریافت می‌کند (چه آنی چه با تأخیر) به عنوان اطلاعات آموزشی حداکثر کند. بازی چکرز که پیش‌تر در فصل ۱ بررسی کردیم نمونه‌ای از همین نوع مسئله است.

۱,۵ خلاصه و منابع برای مطالعه‌ی بیشتر

یادگیری ماشین به سؤالاتی نظیر چگونگی ساخت برنامه‌های کامپیوتری‌ای که بتوانند کارایی‌شان را در انجام کاری بعد از تجربه افزایش دهند می‌پردازد. نکات کلیدی این فصل شامل موارد زیر می‌شود:

- اثبات شده که الگوریتم‌های یادگیری ماشین ارزش عملی زیادی در بسیاری از زمینه‌های کاربردی دارند. این الگوریتم‌ها به طور خاص در (a) مسائل کاوش داده در پایگاه‌های داده‌ای که ممکن است ترتیب‌های محض خاصی را داشته باشد که به صورت اتوماتیک قابل تشخیص هستند (برای مثال، برای بررسی حاصل یک درمان پزشکی بر روی پایگاه داده‌ی بیماران یا یادگیری قوانین کلی بازگشت سرمایه بر روی پایگاه داده‌ی اطلاعاتی)؛ (b) قلمروهایی که انسان دانش کافی برای درک و ارائه‌ی الگوریتم‌های مؤثر در آن‌ها را ندارند (نظیر تشخیص چهره در عکس) و (c) قلمروهایی که برنامه‌ها باید تطبیق پذیر با محیط در حال تغییر باشند (نظیر فرایند تولید با انبار محدود منابع و یا تشخیص علاقه به مطالب برای فردی که علایق متغیری دارد) کاربردهای زیادی دارند.

---

[۱۷] mistake-bound

[۱۸] Weighted majority

- یادگیری ماشین به سمت ایده‌ای از مجموعه‌ی متنوعی از قوانینی شامل هوش مصنوعی، احتمال، آمار، پیچیدگی محاسباتی، تئوری اطلاعات، روانشناسی، تئوری کنترل و فلسفه می‌رود.
- یک مسئله‌ی یادگیری خوش تعریف نیاز به هدف، معیار عملکرد و منبع تجربیات آموزشی ی دقیق تعریف شده[۱۹] دارد.
- طراحی یک روش یادگیری ماشین شامل تعدادی انتخاب‌های طراحی نظیر تعیین نوع تجربیات آموزشی، تابع هدف یادگیری، نمایشی برای این تابع هدف، و الگوریتمی برای یادگیری تابع هدف از تجربیات آموزشی می‌شود.
- یادگیری به نگاه جستجو: جستجو میان فضایی از فرضیه‌های ممکن برای پیدا کردن فرضیه‌ای که بهترین تطابق را با نمونه‌های آموزشی و قیود و دانش اولیه داشته باشد. در اکثر فصول این کتاب بر متدهای یادگیری مختلفی است که فضاهای فرضیه‌ای مختلفی را جستجو می‌کنند تأکید می‌کنیم (برای مثال، فضای توابع عددی یا شبکه‌های عصبی یا درخت‌های تصمیم یا قوانین نمادین یا ...) و نتایج تئوری‌ای در مورد شرایط همگرایی این متدها به فرضیه‌ی بهینه را بررسی خواهیم کرد.

منابع خوبی برای مطالعه درباره‌ی آخرین تحقیقات در یادگیری ماشین وجود دارد. مجله‌های مربوط شامل Machine Learning، Neural Computation، Neural Networks، Journal of the American Statistical Association، IEEE Transacations on Pattern Analysis و Machine Intelligence می‌شوند. همچنین همایش‌های سالانه‌ای که جنبه‌های مختلفی از یادگیری ماشین بر گزار می‌شود، این همایش‌ها نظیر International Conference on Machine Learning، Neural Information Processing Systems، Conference on Computation Learning Theory، International Conference on Genetic Algorithms، International Conference on Knowledge Discovery and Data Mining، European Conference on Machine Learning و دیگر همایش‌ها هستند.

تمارین

۱,۱. به سه کامپیوتر کاربردهایی دهید که به نظر می‌رسد یادگیری ماشین در آن‌ها غیر کاراست و سه کاربردی که به نظر می‌رسد یادگیری ماشین مناسب است. کاربردهایی را انتخاب کنید که در فصل ذکر نشده‌اند و توجیه کوتاهی برای هر کدام بیاورید.

۱,۲. چند کار یادگیری که در فصل آورده نشده‌اند را انتخاب کنید. به طور غیر رسمی این کارها را در پاراگرافی توصیف کنید. حال این کار را با استفاده از تعیین تا حد ممکن دقیق کار، معیار کارایی و تجربیات آموزشی انجام دهید. تابع هدف و نمایشی برای آن تعیین کنید تا بتوان تابع را یاد گرفت. معیارهای اساسی‌ای را که در توصیف دقیق این کار به‌کاربرده این را تعیین کنید.

۱,۳. اثبات کنید که قانون تغییر وزن LMS که در فصل آورده شد از شیب نزول برای رسیدن به مینیمم خطای مربعی استفاده می‌کند. در حالت خاص خطای مربعی E مشابه آنچه در متن فصل آورده شد تعریف می‌شود. حال مشتق E را نسبت به وزن w_i با فرض اینکه $\hat{V}(b)$ تابع خطی تعریف شده در متن درس است حساب کنید. شیب نزول با تغییر هر وزن در جهت $-\frac{\partial E}{\partial w_i}$ است. بنابراین شما باید نشان دهید که قانون آموزش LMS برای هر نمونه‌ی آموزشی وزن‌ها را در این جهت تغییر می‌دهد.

۱,۴. استراتژی‌های مختلفی برای سازنده‌ی تجربه‌ی شکل ۱,۲ در نظر بگیرید. در کل، استراتژی‌هایی را در نظر بگیرید که این سازنده‌ی تجربه چینش‌های صفحه‌ای را ارائه دهد که:

---

[۱۹] well-specified

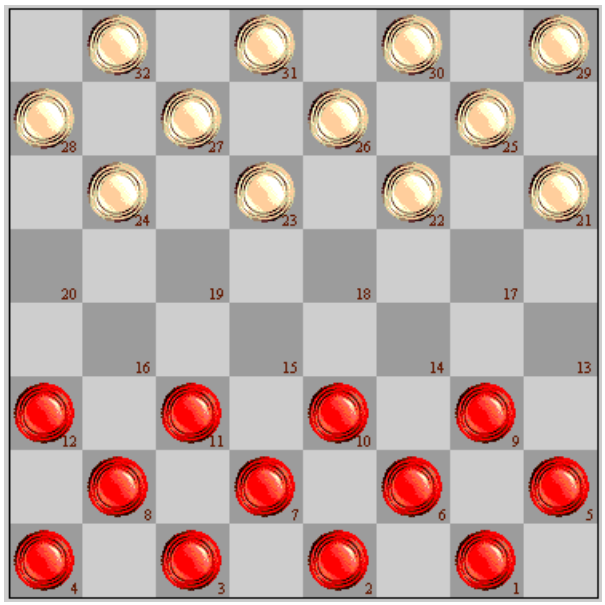
- چینش صفحه تصادفی باشد (اما ممکن[۲۰] باشد)
- چینش صفحه با انتخاب یک چینش صفحه از بازی قبل و اعمال چند حرکت که در آن بازی انجام نشده باشد
- استراتژی دلخواه خودتان

معیارهای این استراتژی‌ها را با هم مقایسه و بررسی کنید. اگر می‌خواستید یکی از این استراتژی‌ها را برای یادگیری با تعداد تجربه‌ی آموزشی ثابت با معیار عملکرد تعداد بیشتر برد در مسابقات جهانی کدام عملکرد بهتری خواهد داشت؟

۱,۵. الگوریتمی مشابه الگوریتم مطرح شده برای مسئله‌ی بازی چکرز را برای بازی دوز[۲۱] ارائه دهید. تابع یاد گرفته شده‌ی $\hat{V}$ را ترکیب خطی ویژگی‌های دلخواه صفحه در نظر بگیرید. برای آموزش مسئله‌ی خود، از بازی برنامه در مقابل نسخه‌ی دومی از همان برنامه که از معیار عملکرد ثابتی که دست‌نویس خودتان باشد استفاده کنید. نمودار درصد بازی‌های برده را بر حسب تعداد بازی انجام شده در مرحله‌ی آموزش را رسم کنید.

۱,۶ بازی چکرز (برای خوانندگانی که با این بازی آشنایی ندارند) (اضافه شده توسط مترجم)

بازی چکرز یک بازی دونفره است که دو طرف، روی یک صفحه‌ی مربع با طول و عرض ۸ خانه و به نوبت با همدیگر بازی می‌کنند. در شکل زیر صفحه‌ای صفحه‌ی بازی چکرز را مشاهده می‌کنید:



[۲۰] legal

[۲۱] tic-tac-toe

علائم و نشانه‌هایی از این بازی را در آثار به جا مانده از مصر باستان مربوط به ۱۶۰۰ سال قبل از میلاد مسیح را دید. بازی‌هایی به این سبک و در شکل‌های گوناگون از زمین، را می‌توان در نقاط گوناگون از دنیا مشاهده کرد.

برخی از قوانین ابتدایی این بازی به این شکل است:

- هر بازیکن دارای ۱۲ مهره است که در خانه‌های تیره چیده شده‌اند.
- مهره‌های هر فرد فقط می‌توانند بر روی خانه‌های تیره حرکت کنند.
- حرکات به صورت ضربدری انجام می‌شود.
- اگر مهره‌ی حریف در مسیر حرکت قرار گیرد اگر خانه‌ی بعدی خالی باشد می‌توان با پرش از روی مهره‌ی حریف آن مهره را زد. اگر بازیکنی بتواند مهره‌ی حریف را بزند نمی‌تواند حرکت دیگری انجام دهد.
- می‌توان در یک حرکت بیش از یک مهره‌ی حریف را به صورت پشت سر هم زد.

فرهنگ لغات تخصصی فصل (فارسی به انگلیسی)

تجربه	experience
تشخیص دستخط	recogonize handwritten words
خوش وضع	well-posed
کارایی	performance
یادگیری	learning
کار	Task