

هوش مصنوعی، یادگیری ماشین کاربرد موردی: پردازش گفتار

مدرس: کیان ابراهیم کافوری
دانشگاه آزاد اسلامی واحد یادگار امام خمینی (ره) شهرری

هوش مصنوعی، یادگیری ماشین کاربرد موردی: پردازش گفتار

• مدرس: کیان ابراهیم کافوری

- email: kian_kafoori@yahoo.com
- web page: <http://kafoori.blog.ir>

• مطالب دوره:

1. بخش هوش مصنوعی و یادگیری ماشین

2. بخش ابزار یادگیری ماشین (PyTorch)

3. بخش پردازش گفتار

هوش مصنوعی، یادگیری ماشین کاربرد موردی: پردازش گفتار

• ارزشیابی درس:

- امتحان پایان ترم (۱۴ نمره) + ارائه، پروژه و امتحان شفاهی کلاسی (۶ نمره).
- دانشجو باید بر مبنای مقاله ای مرتبط با درس از مجلات معتبر و به روز (دو سال اخیر) ارائه ای را آماده سازی کند که در آخرین جلسات کلاس برگزار خواهد شد.
- مشخصات پروژه در طی کلاس ارائه می شود.
- حضور در کلاس الزامی است.

منابع و مراجع درس

• بخش هوش مصنوعی و یادگیری ماشین:
• کتاب:

- **Artificial Intelligence: A Modern Approach, 2020**, by Stuart Russell , Peter Norvig

• دوره:

- **Artificial Intelligence: Principles and Techniques**, Stanford ,2024, online at stanford-cs221.github.io

• بخش ابزار یادگیری ماشین (PyTorch)
• کتاب:

- **Deep Learning with PyTorch, 2020**, BY Eli Stevens, Luca Antiga, Thomas Viehmann

• دوره:

- **Learn PyTorch for Deep Learning: Zero to Mastery book**, online at www.learnpytorch.io

• بخش پردازش گفتار
• دوره ها:

- **Introduction to Speech Processing**, [Aalto University](https://aalto.fi), Finland, online at speechprocessingbook.aalto.fi
- **Fundamentals of Speech and Language Processing**, CUHK, Hong Kong, online at slpcourse.github.io

• بخش کمکی (آموزش زبان برنامه نویسی Python)

- [Learnpython.org](https://learnpython.org), learn-python.adamemery.dev

مطالب درس

- بخش هوش مصنوعی و یادگیری ماشین:

- هوش مصنوعی:

- مقدمه (تاریخچه و اصول هوش مصنوعی)
- سه بنیان الگوی اساسی هوش مصنوعی (مدل سازی، نتیجه گیری، یادگیری)

- یادگیری ماشین:

- مدل های براساس Reflex
- مدل های براساس State
- مدل های براساس Variable
- منطق

مطالب درس

- بخش ابزار یادگیری ماشین (PyTorch):

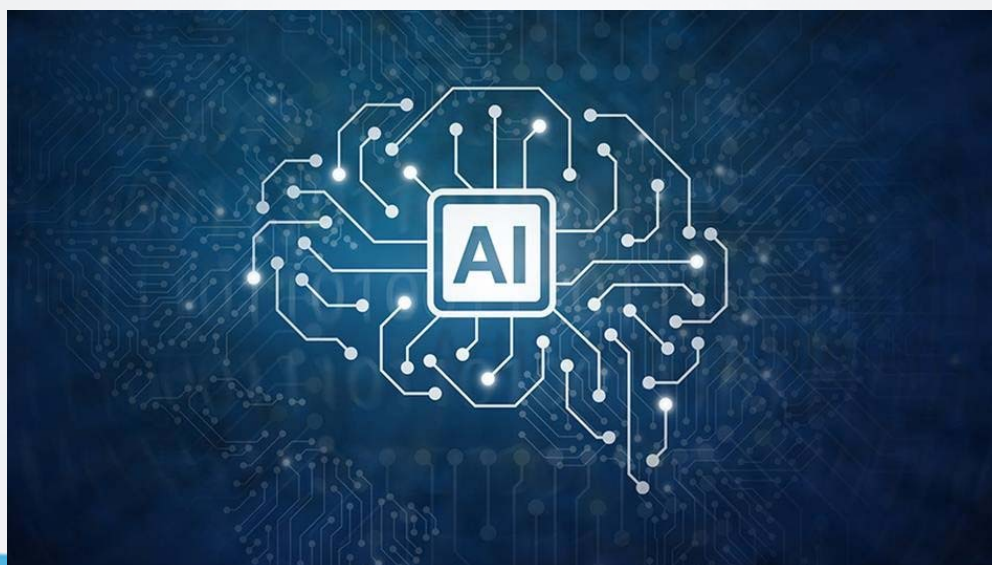
- اصول PyTorch
- روش کار با PyTorch
- طبقه بندی شبکه عصبی با PyTorch
- مثال بینایی ماشین با PyTorch
- ویرایش و استفاده از داده ها با PyTorch
- ماژولار بودن در PyTorch
- یادگیری انتقالی در PyTorch
- اجرای یک پروژه با PyTorch

مطالب درس

• بخش پردازش گفتار

- مقدمه
- نمایش‌های پایه گفتار
- پیش‌پردازش گفتار
- ابزارهای مدل‌سازی در پردازش گفتار
- ارزیابی روش‌های پردازش گفتار
- تحلیل گفتار
- بازشناسی‌های مختلف در گفتار
- تولید گفتار
- بهسازی گفتار

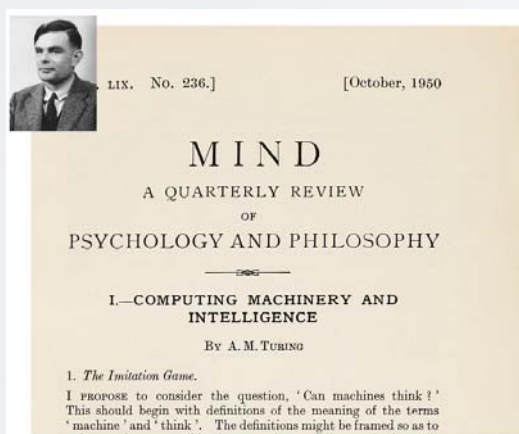
هوش مصنوعی



پیشرفت هوش مصنوعی در سال‌های اخیر

- طور کلی این روزها به ایجاد انگیزه در هوش مصنوعی دشوار نیست.
- هوش مصنوعی در خیلی از بازی‌ها استادان انسانی را شکست داده است.
- Jeopardy! (IBM Watson, 2011), Go (DeepMind's AlphaGo, 2016), Dota 2 (OpenAI, 2019), Poker (CMU and Facebook, 2019)
- در کارهای غیر بازی، سیستم‌هایی داریم که عملکرد قوی در درک مطلب، بازشناسی گفتار، تشخیص چهره و معیارهای تصویربرداری پزشکی دارند.
- با این حال، برخلاف بازی‌ها، که هدفی مشخص دارد (پیروزی بر رقیب)، عملکرد خوب در یک معیار لزوماً به عملکرد خوب در کار واقعی در طبیعت تبدیل نمی‌شود.
- صرفاً به این دلیل که شما در یک معیار خوب جواب می‌گیرید، لزوماً به این معنی نیست که درک کاملی دارید یا می‌دانید چگونه آن دانش را در مسائل واقعی به کار ببرید. بنابراین، اگرچه امیدوارکننده است، اما همه این نتایج به برنامه‌های کاربردی در دنیای واقعی منجر نمی‌شوند.

تاریخچه هوش مصنوعی



- اولین مورد تاریخچه هوش مصنوعی مقاله برجسته آلن تورینگ است که در هوش منتشر شد. در این مقاله، تورینگ این سوال را مطرح کرد که imitation game که بیشتر به عنوان تست تورینگ شناخته می‌شود به
- ماشینی تست تورینگ را می‌گذراند که بتواند یک قاضی انسانی را متقاعد گفتگوی زبان طبیعی پاسخگو است.
- این مقاله قابل توجه است نه به این دلیل که سیستمی را ساخته یا روشی بحث‌های فلسفی را برای دهه‌های آینده چارچوب می‌دهد.
- درک مفهومی مانند هوش دشوار است. بنابراین این واقعاً اولین پاسخ رسمی می‌تواند فکر کنند؟ اینکه آیا قبولی در آزمون تورینگ چیزی است که با بحث برانگیز است، اما پیامدهای فلسفی آن کاملاً تأمل برانگیز است.
- یکی از نکات مهم آزمون تورینگ، جداسازی مشخصات عینی کاری است که ما می‌خواهیم یک سیستم انجام دهد ("چه") از روش‌هایی که ممکن است ما را به آنجا برساند ("چگونه"). این جداسازی یک موضوع اصلی است.
- در پایان مقاله، تورینگ دو رویکرد ممکن را مورد بحث قرار می‌دهد. اولین مورد مبتنی بر حل مسائل انتزاعی مانند شطرنج است که مسیری است که هوش مصنوعی نمادین طی می‌کند. مورد دوم جایی است که یک ماشین می‌سازید و مانند یک کودک آموزش می‌دهید، این مسیری است که توسط هوش مصنوعی عصبی و آماری طی می‌شود.

تولد AI و موفقیت های اولیه

• سال ۱۹۵۶:

- شرکت کنندگان یک کنفرانس پیشنهاد جسورانه ای ارائه کردند: ایجاد سیستمی که بتواند هر جنبه ای از هوش را به تصویر بکشد. آنها دنبال Generality بودند.
- در واقع، در این دوران پس از جنگ، کامپیوترها تازه وارد صحنه شدند. زمان بسیار هیجان انگیزی بود و محققان جاه طلب بودند.

- چکرز (۱۹۵۲): برنامه ساموئل بازی چکرز را یاد گرفت و در سطح آماتوری قوی بازی کرد
- حل مسئله (۱۹۵۵): نظریه پرداز منطقی: اثبات قضایای در ریاضیات با استفاده از جستجو + اکتشاف. بعداً، حل المسائل عمومی (GPS)

رشد AI در طول تاریخ

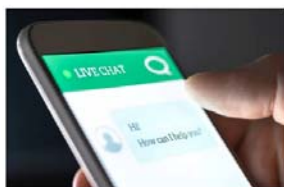
- خوشبینی های بیش از حد اولیه:
- ماشین ها ظرف بیست سال قادر خواهند بود هر کاری را که یک انسان می تواند انجام دهد، انجام دهند. - هربرت سایمون
- در عرض ۱۰ سال مشکلات هوش مصنوعی به طور قابل توجهی حل خواهد شد. - ماروین مینسکی
- زمانی را تجسم می کنم که برای ربات ها همان چیزی خواهیم بود که سگ ها برای انسان ها هستند. - کلود شانون
- نتایج ناامیدکننده اولیه:
- علیرغم موفقیت ها، برخی وظایف مانند ترجمه ماشینی با شکست کامل مواجه شدند.
- با این حال، این ترجمه برای سازمان های دولتی که بودجه تحقیقات را تامین می کنند چندان سرگرم کننده نبود.
- در سال ۱۹۶۶، گزارش ALPAC منجر به قطع بودجه برای ترجمه ماشینی شد.
- این آغاز اولین زمستان هوش مصنوعی بود.

پیامدهای دوران اولیه مشکلات:

- محاسبات محدود: فضای جستجو به طور تصاعدی رشد کرد و از سخت افزار پیشی گرفت.
- اطلاعات محدود: پیچیدگی مشکلات هوش مصنوعی (تعداد کلمات، اشیاء، مفاهیم در جهان)

مشکلات پیچیده دنیای واقعی

- هوش مصنوعی در مورد استفاده از فناوری برای مقابله با مشکلات پیچیده دنیای واقعی است.
- برخی از مشکلات پیچیده دنیای واقعی که به ذهن می رسد چیست؟
- خودروهای خودران باید صحنه پیچیده روبروی خود را حس کنند (در زمان واقعی!) و تصمیمات ایمن بگیرند که خودرو را به سمت مقصد پیش می برد.
- دستیاران مجازی باید ظرافت های غنی زبان طبیعی را درک کنند، قصد کاربر را استنباط کنند و پاسخ های مناسب را ایجاد کنند.
- سیستم های پزشکی باید نشانه های ظریف و نویزی را در تصاویر پزشکی تفسیر کنند، اطلاعات ناقص بیمار را در نظر بگیرند و تشخیص های قابل اعتمادی را ایجاد کنند که قابل اعتماد پزشکان باشد.
- اینها واقعاً مسائل سختی هستند. اصلاً چگونه شروع به مقابله با آنها می کنید؟



پُر کردن شکاف



- در نهایت، باید مقداری کد بنویسیم (و احتمالاً سخت افزاری نیز بسازیم).
- اما شکاف بزرگی بین مشکل و راه حل وجود دارد.
- شما نمی خواهید فقط شروع به پریدن و نوشتن کد بدون برنامه ریزی واضح برای حمله کنید.

روش کار

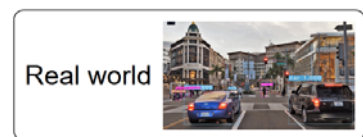
Modeling

Inference

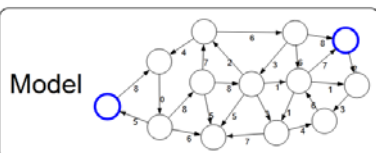
Learning

- بنابراین در اینجا طرح حمله است. ما الگوی یادگیری مدلسازی-استنتاج را برای کمک به حرکت در فضای راه حل اتخاذ خواهیم کرد.
- سه رکن وجود دارد، مدل سازی، استنتاج و یادگیری.
- در واقعیت، خطوط بین سه ستون مبهم است، اما این روش کار به عنوان یک اصل راهنمای ایده آل و مفید عمل می کند.

مدل سازی

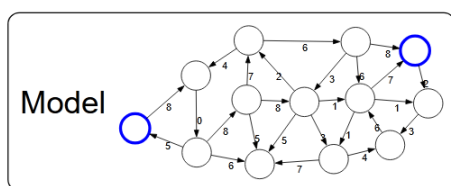


Modeling

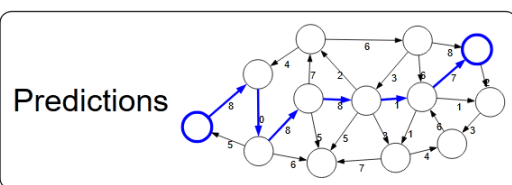


- رکن اول مدل سازی است.
- مدل سازی مسائل آشفته دنیای واقعی را می گیرد و آن ها را در اشیاء ریاضی رسمی و منظمی به نام مدل ها بسته بندی می کند که می توانند در معرض تجزیه و تحلیل دقیق قرار گیرند و توسط رایانه ها قابل اجرا باشند.
- با این حال، مدل سازی با اتلاف (LOSSY) است: نمی توان تمام غنای دنیای واقعی را ایجاد کرد، و بنابراین یک هنر مدل سازی وجود دارد: چه چیزی را در مقابل نادیده گرفتن آن نگاه می داریم؟
- (یک استثناء در این مورد، بازی هایی مانند شطرنج، برو یا سودوکو هستند که در آن دنیای واقعی مشابه مدل است.)
- ما ممکن است مشکل رانندگی را به عنوان یک مسئله یافتن مسیر به عنوان یک نمودار فرموله کنیم که در آن گره ها نقاط شهر را نشان می دهند، یال ها نشان دهنده جاده ها و هزینه یک یال نشان دهنده ترافیک آن جاده است.
- اگر این کار را انجام دهیم، تمام پیچیدگی های ادراک نادیده گرفته می شود. از طرف دیگر، می توانیم مدلی بسازیم که فقط به ادراک نگاه می کند و برنامه ریزی را نادیده می گیرد.
- شایان ذکر است که مدل سازی با اتلاف می تواند منجر به خطاهایی در سیستم هوش مصنوعی شود که می تواند تأثیر نامطلوبی بر افراد داشته باشد. بخش عمده ای از توسعه هوش مصنوعی مسئول درک و کاهش این تأثیر است.

استنتاج

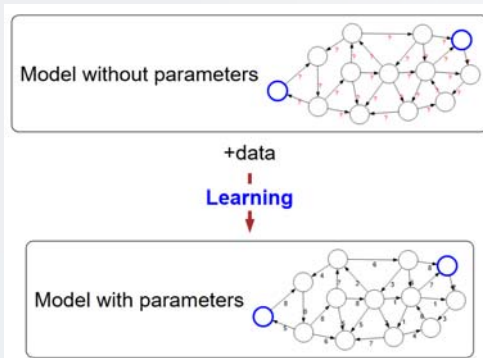


Inference



- رکن دوم استنتاج است.
- با توجه به یک مدل، وظیفه استنتاج پاسخ دادن به سؤالات با توجه به مدل است. به عنوان مثال با توجه به مدل شهر می توان سؤالاتی از این قبیل پرسید: کوتاه ترین مسیر کدام است؟ ارزان ترین مسیر چیست؟
- تمرکز استنتاج معمولاً روی الگوریتم های کارآمدی است که می توانند به این سؤالات پاسخ دهند.
- برای برخی از مدل ها، پیچیدگی محاسباتی می تواند یک نگرانی باشد (بازی هایی مانند GO) و معمولاً به تقریب هایی نیاز است.

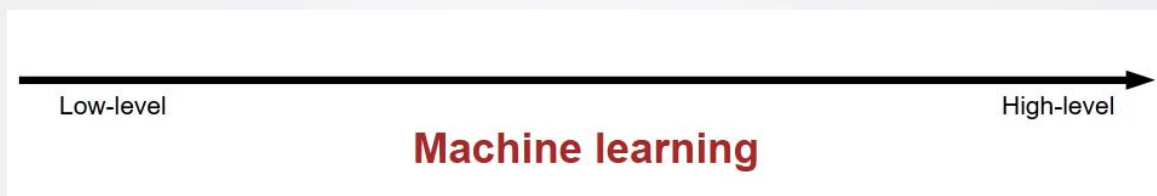
یادگیری



- اما این مدل از کجا آمده است؟ به یاد داشته باشید که دنیای واقعی پیچیده است، بنابراین اگر الگو قرار است وفادار باشد، مدل نیز باید پیچیده باشد. اما ما نمی‌توانیم چنین مدل غنی را به صورت دستی ایجاد کنیم.
- ایده پشت یادگیری (ماشینی) این است که در عوض آن را از داده‌ها دریافت کنید. به جای ساختن یک مدل، یک اسکلت از یک مدل (به طور دقیق‌تر، یک خانواده مدل) ساخته می‌شود، که یک مدل بدون پارامتر است. و سپس اگر داده‌های مناسب را داشته باشیم، می‌توانیم یک الگوریتم یادگیری ماشینی را برای تنظیم پارامترهای مدل اجرا کنیم.
- احتمالاً بسیاری از شما فقط مدل‌هایی را در زمینه یادگیری ماشینی مشاهده کرده‌اید (مثلاً شبکه‌های عصبی)، اما من می‌خواهم همه شما دیدگاه بسیار گسترده‌ای نسبت به اینکه مدل چیست داشته باشید. ایده یادگیری به یک خانواده مدل خاص (به عنوان مثال، شبکه‌های عصبی) وابسته نیست. بلکه بیشتر فلسفه چگونگی تولید مدل است.

خلاصه روش کار

- به طور خلاصه، مدل‌سازی دنیای واقعی را ساده می‌کند، استنتاج به سؤالات در برابر مدل پاسخ می‌دهد، و یادگیری مدل را از داده‌ها می‌سازد.
- توجه داشته باشید که انجام هر مرحله می‌تواند چالش برانگیز باشد و نیاز به تقریب داشته باشد. احتمالاً trade-off هایی نیز وجود خواهد داشت.



- اکنون در این دوره، انواع مختلفی از مدل‌ها را برای نمایش انواع مختلف مسائل دنیای واقعی مرور می‌کنیم.
- ما با مدل‌هایی شروع می‌کنیم که سطح پایین هستند (که فقط یک ورودی دریافت می‌کنند و یک خروجی برمی‌گردانند) - اینها مدل‌هایی هستند که ممکن است در یادگیری ماشینی با آنها بیشتر آشنا شوید). به تدریج به سمت مدل‌های سطح بالا که مبتنی بر استدلال منطقی تر هستند پیش می‌رویم.
- قبل از شروع، ما در مورد یادگیری ماشین صحبت می‌کنیم که می‌تواند از همه مدل‌ها پشتیبانی کند.



یادگیری ماشین

- محرک اصلی موفقیت‌های اخیر در هوش مصنوعی

- انتقال پیچیدگی از "کد" به "داده"

- نیاز به یک توانایی بزرگ: تعمیم

- پشتیبانی از همه این مدل‌ها، یادگیری ماشین است، که مسلماً مهم‌ترین عنصری است که موفقیت‌های اخیر در هوش مصنوعی را تقویت کرده است. از منظر عملی، یادگیری ماشینی به ما امکان می‌دهد پیچیدگی مدل را از کد به داده تغییر دهیم. به جای نوشتن کدهای زیادی برای توصیف مدل، در عوض یک خانواده مدل بسیار ساده (مثلاً انتخاب معماری شبکه عصبی) می‌نویسیم و بر جمع‌آوری داده‌ها برای آموزش یک مدل در آن خانواده تمرکز می‌کنیم.
- بخش جادویی مفهومی اصلی یادگیری این است که اگر به درستی انجام شود، مدل آموزش دیده می‌تواند پیش‌بینی‌های خوبی فراتر از مجموعه نمونه‌های آموزشی ایجاد کند. این توانایی بزرگ، تعمیم نامیده می‌شود و به طور صریح یا ضمنی، در قلب هر الگوریتم یادگیری ماشینی قرار دارد. این را می‌توان با استفاده از ابزارهای احتمالات و نظریه یادگیری آماری رسمی کرد.

مدل های مبتنی بر رفلکس

• مثال ها: طبقه بندی کننده های خطی، شبکه های عصبی عمیق

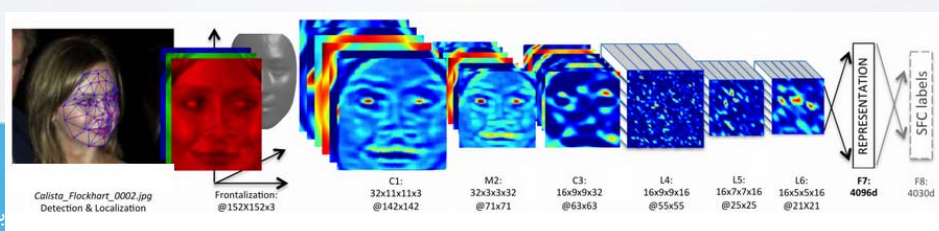
• رایج ترین مدل ها در یادگیری ماشینی

• کاملاً فید فوروارد (بدون پس گرد)

• یک مدل مبتنی بر رفلکس به سادگی یک توالی ثابت از محاسبات را بر روی یک ورودی مشخص انجام می دهد. مثال ها شامل اکثر مدل هایی است که در یادگیری ماشینی یافت می شوند، از طبقه بندی کننده های خطی ساده تا شبکه های عصبی عمیق.

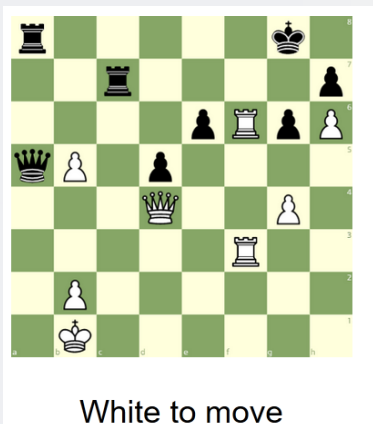
• مشخصه اصلی مدل های مبتنی بر رفلکس این است که محاسبات آن ها پیشخور است (از ورودی به خروجی). کسی عقب نشینی نمی کند و محاسبات جایگزین را در نظر می گیرد.

• استنباط در این مدل ها ساده است زیرا فقط محاسبات ثابت را اجرا می کند که این مدل ها را جالب می کند.



پراهمیم کافوری

23



White to move

مدل های مبتنی بر حالت

• این وظیفه را در نظر بگیرید که سفید با توجه به موقعیت شطرنج خاص چه حرکتی باید انجام دهد. بسیاری از ما این را دشوارتر از شناخت صاحب عکس خواهیم یافت.

• برنامه های کاربردی:

• بازی ها: شطرنج، Go، Pac-Man، Starcraft و غیره.

• رباتیک: برنامه ریزی حرکت

• مدل های مبتنی بر رفلکس برای کارهایی که نیاز به دقت بیشتری دارند (مثلاً در بازی شطرنج یا برنامه ریزی یک سفر بزرگ) بسیار ساده هستند. مدل های مبتنی بر حالت بر این محدودیت غلبه می کنند.

• ایده کلیدی، در سطح بالا، مدل سازی وضعیت یک جهان و انتقال بین حالت هایی است که توسط اقدامات ایجاد می شود. به طور مشخص، می توان حالت ها را به عنوان گره ها در نمودار و انتقال ها را به عنوان یال ها در نظر گرفت. این کاهش مفید است زیرا ما نمودارها را به خوبی درک می کنیم و الگوریتم های کارآمد زیادی برای کار بر روی نمودارها داریم.

مدل های مبتنی بر حالت

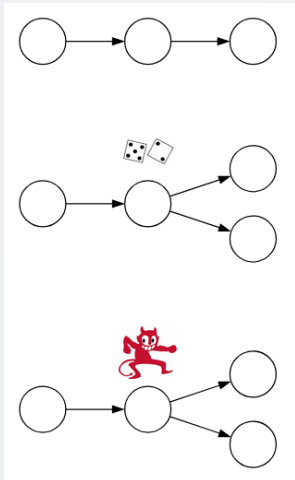
• مسائل جستجو: شما همه چیز را کنترل می کنید

• فرآیندهای تصمیم مارکوف: در برابر طبیعت (بازی های داری تاس)

• بازی های خصمانه: در برابر حریف (به عنوان مثال، شطرنج)

• ما سه نوع مدل مبتنی بر حالت را در نظر می گیریم:

- مسائل جستجو مدل هایی هستند که در آنها در محیطی کار می کنید که هیچ ابهامی ندارد. با این حال، در بسیاری از موقعیت های واقع گرایانه، نیروهای دیگری موثر هستند
- فرآیندهای تصمیم مارکوف موقعیت هایی را که در آن تصادفی وجود دارد (به عنوان مثال، بلک جک) کنترل می کند.
- بازی های خصمانه، همانطور که از نام آن پیداست، وظایفی را انجام می دهند که در آن حریفی وجود دارد که علیه شما کار می کند (به عنوان مثال، شطرنج).



مدل های مبتنی بر متغیر

• مثال: بازی سودوکو

• هدف: ارقام را در مربع های خالی قرار دهید تا هر سطر، ستون و بلوک فرعی ۳*۳ دارای ارقام ۱-۹ باشد.

• کلید: ترتیب پر کردن مربع ها در معیارهای ارزیابی مهم نیست!

• در مدل های مبتنی بر حالت، راه حل ها رویه ای هستند: آنها دستورالعمل های گام به گام را در مورد چگونگی رفتن از A به B مشخص می کنند.

• در برخی از کاربردها، ترتیب انجام کارها مهم نیست.

• به عنوان مثال، در سودوکو، جایی که هدف قرار دادن ارقام در مربع های خالی برای برآوردن برخی محدودیت ها است، تنها چیزی که اهمیت دارد پیکربندی نهایی اعداد است. می توانید آنها را به هر ترتیبی پر کنید.

• مدل های مبتنی بر متغیر به شما این امکان را می دهند که بخواهید درخواست خود را اعلام کنید (مثل یک زبان سطح بالاتر) به جای اینکه مدیریت خرد نحوه یافتن راه حل را انجام دهید.



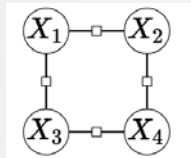
5	3		7						
6			1	9	5				
	9	8					6		
8			6					3	
4			8	3				1	
7				2					6
	6					2	8		
			4	1	9			5	
			8			7	9		



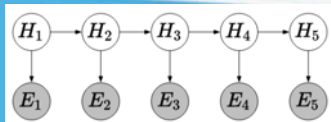
5	3	4	6	7	8	9	1	2	
6	7	2	1	9	5	3	4	8	
1	9	8	3	4	2	5	6	7	
8	5	9	7	6	1	4	2	3	
4	2	6	8	5	3	7	9	1	
7	1	3	9	2	4	8	5	6	
9	6	1	5	3	7	2	8	4	
2	8	7	4	1	9	6	3	5	
3	4	5	2	8	6	1	7	9	

مدل‌های مبتنی بر متغیر

- مسائل رعایت محدودیت: محدودیت‌های سخت (مانند سودوکو، زمان‌بندی)
- شبکه‌های بیزی: وابستگی‌های نرم (به عنوان مثال، ردیابی اتومبیل‌ها از حسگرها)
- مسایل رعایت محدودیت: مدل‌های مبتنی بر متغیر هستند که در آنها فقط محدودیت‌های سخت داریم. مثلاً در برنامه ریزی یک نفر نمی‌تواند همزمان در دو مکان باشد.



- شبکه‌های بیزی مدل‌های مبتنی بر متغیر هستند که در آن متغیرها متغیرهای تصادفی هستند که به یکدیگر وابسته هستند. به عنوان مثال، مکان واقعی یک هواپیما H_t و خوانش راداری آن E_t ، و همچنین مکان H_t به مکان در آخرین مرحله زمانی H_{t-1} مرتبط هستند. ساختار وابستگی دقیق توسط نمودار داده شده است و به طور رسمی توزیع احتمال مشترک را بر روی همه متغیرها تعریف می‌کند.



هوش مصنوعی، یادگیری ماشین، کاربرد موردی: پردازش گفتار

27

منطق: انگیزه: دستیار مجازی

- دستیار مجازی باید:

- اطلاعات ناهمگن را هضم کند.
- با این اطلاعات عمیقاً استدلال کند.

- یکی از انگیزه‌های منطق، دستیار مجازی است. یک دستیار خوب باید بتواند آنچه را که به او گفتید به خاطر بسپارد و به سؤالاتی که مستلزم استنباط از دانش خود است پاسخ دهد. و شما می‌خواهید یا استفاده از زبان طبیعی، ابزاری که انسان‌ها برای برقراری ارتباط اختراع کردند، با آن تعامل کنید.



- تعامل با این سیستم بسیار متفاوت از یک سیستم معمولی مبتنی بر یادگیری ماشین است. اول، تطبیق پذیر است، در حالی که اکثر سیستم‌های ML یک عملکرد ثابت هستند. دوم، انواع اطلاعاتی که ما در اختیار آنها قرار می‌دهیم و انواع سؤالات ناهمگون تر و انتزاعی تر هستند و ما انتظار داریم که سیستم به پیامدهای کامل تک تک کلمات پی ببرد. (ما نمی‌خواهیم ۱۰۰ بار به دستیار شخصی خود بگوییم که جلسات صبحگاهی را ترجیح می‌دهیم.)

- اخیراً، با ظهور مدل‌های زبان بزرگی مانند ChatGPT، ما شروع به دیدن دستیارهای مجازی مانند نمونه‌ای که در اینجا نمایش داده شده است، اما با قابلیت‌های بسیار بیشتر، شروع شده‌ایم. با این حال، LLMها به دلیل توهمات خود شناخته شده هستند، در حالی که سیستم مبتنی بر منطق ۱۰۰٪ از درون سازگار است.

- اغلب هوش مصنوعی منطقی و هوش مصنوعی آماری را در تضاد قرار می‌دهند. در این دوره، این دو را نه متضاد بلکه مکمل یکدیگر می‌دانیم. منطق دسته‌ای از مدل‌ها را ارائه می‌کند که سطح بالاتری دارند، اما همچنان باید توسط داده‌های واقعی که یادگیری ماشین ارائه می‌دهد، پشتیبانی شوند.

خلاصه مباحث هوش مصنوعی

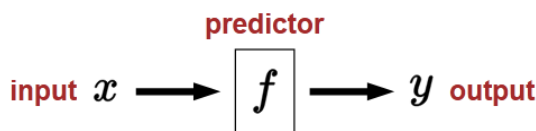
- به طور خلاصه، مدل‌ها را مورد بحث قرار می‌دهیم، از مدل‌های مبتنی بر رفلکس به مدل‌های مبتنی بر حالت و مدل‌های مبتنی بر متغیر و منطق. برای هر کدام، ما الگوی یادگیری مدلسازی-استنتاج-یادگیری را مثال می‌زنیم.
- در این درس هدف بیشتر به مدل‌های مبتنی بر رفلکس می‌پردازیم.

یادگیری ماشین

بررسی اجمالی

یادگیری ماشین

- به یاد بیاورید که یادگیری ماشینی فرآیند تبدیل داده ها به مدل است. سپس با آن مدل می توانید استنتاجی را روی آن انجام دهید تا پیش بینی کنید.
- در حالی که یادگیری ماشین می تواند برای هر نوع مدلی اعمال شود، ما توجه خود را بر روی مدل های مبتنی بر بازتاب (رفلکس) متمرکز خواهیم کرد، که شامل مدل هایی مانند طبقه بندی کننده های خطی و شبکه های عصبی است. در مدل های مبتنی بر رفلکس، استنتاج (پیش بینی) شامل مجموعه ای ثابت از عملیات سریع و پیش خور است.
- **مدل های مبتنی بر رفلکس:**
 - به طور انتزاعی، یک مدل مبتنی بر رفلکس (که ما آن را پیش بین می نامیم f) مقداری ورودی x می گیرد و مقداری خروجی y تولید می کند.
 - ورودی معمولاً می تواند دلخواه باشد (یک تصویر یا جمله)، اما شکل خروجی معمولاً محدود است و نوع کار پیش بینی را تعیین می کند.



هوش مصنوعی، یادگیری ماشین، کاربرد موردی: پر

31

طبقه بندی باینری

- نمونه کاردها:

- تشخیص تقلب: تقلب در تراکنش کارت اعتباری یا عدم تقلب
- کامنت های ناهنجار: کامنت آنلاین ناهنجار یا مناسب است
- بوزون هیگز: اندازه گیری رویداد یا پس زمینه فروپاشی رویداد

- گسترش: طبقه بندی چند کلاسه:

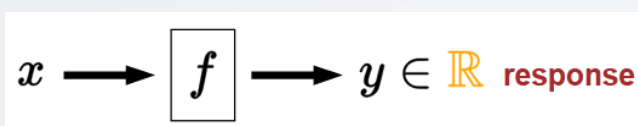
$$y \in \{1, \dots, K\}$$

- یکی از کارهای رایج پیش بینی، طبقه بندی باینری است، که در آن خروجی معمولاً به صورت مثبت (+1) یا منفی (-1) بیان می شود.
- در زمینه وظایف طبقه بندی، f طبقه بندی کننده نامیده می شود و y برچسب (گاهی اوقات کلاس، دسته یا برچسب) نامیده می شود.
- در اینجا چند کاربرد عملی آورده شده است.
- یکی از کاربردها تشخیص کلاهبرداری است: با توجه به اطلاعات مربوط به تراکنش با کارت اعتباری، پیش بینی کنید که آیا این تراکنش تقلبی است یا خیر، تا بتوان تراکنش را مسدود کرد

طبقه بندی باینری

- برنامه دیگر تعدیل انجمن های گفتگوی آنلاین است: با توجه به نظر آنلاین، پیش بینی کنید که آیا ناهنجار است (و بنابراین باید پرچم گذاری شود یا حذف شود) یا خیر.
- کاربرد نهایی از فیزیک می آید: پس از کشف بوزون هیگز، دانشمندان به چگونگی تجزیه آن علاقه مند شدند. برخورد دهنده بزرگ هادرون در سرن پروتون ها را به یکدیگر می کوبد و سپس رویدادهای بعدی را تشخیص می دهد. هدف این است که پیش بینی کنیم که آیا هر رویداد یک بوزون هیگز در حال تجزیه (به دو ذره تاو) است یا فقط نویز پس زمینه است.
- هر یک از این برنامه ها دارای یک مجموعه داده Kaggle هستند.
- همچنین، طبقه بندی چند کلاسه تعمیم طبقه بندی باینری است که در آن خروجی می تواند یکی از K مقدار ممکن باشد. به عنوان مثال، در طبقه بندی رقمی، $K=10$.

رگرسیون Regression



• مثال ها:

- نقشه برداری فقر: تصاویر ماهواره ای
- مسکن: اطلاعاتی در مورد خانه
- زمان ورود: مقصد، آب و هوا، ساعت و تاریخ
- شاخص ثروت و دارایی
- قیمت خانه
- زمان رسیدن

- دومین نوع اصلی از کار پیش بینی که به آن خواهیم پرداخت، رگرسیون است. در اینجا، خروجی y یک عدد حقیقی است (که اغلب پاسخ یا هدف نامیده می شود).
- یکی از کاربردها نقشه برداری فقر است: با توجه به یک تصویر ماهواره ای، میانگین شاخص ثروت دارایی خانه های آن منطقه را پیش بینی کنید. این برای اندازه گیری فقر در سراسر جهان و تعیین مناطقی که بیشترین نیاز را به کمک دارند استفاده می شود.
- کاربرد دیگر: با دادن اطلاعات در مورد یک خانه (به عنوان مثال، مکان، تعداد اتاق خواب)، قیمت آن را پیش بینی کنید.
- سومین کاربرد، پیش بینی زمان رسیدن برخی از خدمات است که می تواند تحویل بسته، پرواز یا اشتراک گذاری باشد.
- مایز کلیدی بین طبقه بندی و رگرسیون این است که طبقه بندی خروجی های مجزا دارد (به عنوان مثال، "بله" یا "خیر" برای طبقه بندی باینری)، در حالی که رگرسیون خروجی های پیوسته دارد.

$$x \longrightarrow \boxed{f} \longrightarrow y \text{ is a complex object}$$

پیش بینی ساختار یافته Structured prediction

• مثال ها:

- ترجمه ماشینی: جمله انگلیسی به جمله ژاپنی
- گفتگو: تاریخچه مکالمه به بیان بعدی
- شرح تصویر: جمله تصویری که تصویر را توصیف می کند
- تقسیم بندی تصویر: تقسیم بندی تصویر

• نوع نهایی کار پیش بینی که در نظر خواهیم گرفت، پیش بینی ساختار یافته است که تا حدودی فراگیر است.

• در پیش بینی ساختار یافته، خروجی y یک شی پیچیده است که می تواند یک جمله یا یک تصویر باشد. بنابراین فضای خروجی های ممکن بسیار زیاد است.

پیش بینی ساختار یافته Structured prediction

- یکی از کاربردها ترجمه ماشینی است: با توجه به یک جمله ورودی در یک زبان، ترجمه آن را به زبان دیگر پیش بینی کنید.
- دیالوگ را می توان به عنوان پیش بینی ساختار یافته مطرح کرد: با توجه به سابقه مکالمه گذشته بین یک کاربر و یک عامل (در مورد دستیاران مجازی)، گفته بعدی را پیش بینی کنید (آنچه نماینده باید بگوید).
- در زیرنویس تصویر، برای فناوری های کمکی بصری بگویید: با توجه به یک تصویر، جمله ای را پیش بینی کنید که آنچه در آن تصویر است را توصیف کند.
- در تقسیم بندی تصویر، که برای بومی سازی اشیاء برای رانندگی خودکار لازم است: با توجه به تصویری از یک صحنه، تقسیم بندی آن تصویر را به مناطق مربوط به اشیاء پیش بینی کنید.
- ایجاد یک تصویر یا یک جمله می تواند مشکل به نظر برسد، اما رازی در اینجا وجود دارد. یک کار پیش بینی ساختار یافته اغلب می تواند به دنباله ای از وظایف طبقه بندی چند طبقه تقسیم شود. به عنوان مثال، برای پیش بینی یک جمله کامل، هر بار یک کلمه را پیش بینی کنید و از چپ به راست بروید. این یک کاهش بسیار قدرتمند است!
- البته: یکی از چالش های این رویکرد این است که خطاها ممکن است آشکار شوند: اگر شروع به اشتباه کردن کنید، ممکن است از ریل خارج شوید و شروع به اشتباهات بیشتر کنید.

مطالب مورد نظر در حوزه یادگیری ماشین

Roadmap

Tasks

Linear regression
Linear classification
K-means

Algorithms

Stochastic gradient descent
Backpropagation

Models

Non-linear features
Feature templates
Neural networks
Differentiable programming

Considerations

Group DRO
Generalization
Best practices

مطالب مورد نظر در حوزه یادگیری ماشین

- ما با صحبت در مورد رگرسیون و طبقه بندی باینری، دو وظیفه اساسی در یادگیری ماشین، شروع خواهیم کرد. به طور خاص، ما ساده ترین تنظیمات را مطالعه می کنیم: **رگرسیون خطی و طبقه بندی خطی**، که در آن مدل های خطی داریم که با نزول گرادیان (Gradient Descent) آموزش داده شده اند.
- در مرحله بعد، ما **نزول گرادیان تصادفی (SGD)** را معرفی می کنیم و نشان می دهیم که می تواند بسیار سریعتر از نزول گرادیان عادی باشد.
- سپس نگاهی دقیق به خطاهای یک مدل می اندازیم و **گروه DRO** را مورد بحث قرار می دهیم، تکنیکی که مطمئن می شود خطاها به طور نابرابر بر روی گروه های مختلف جمعیت قرار نمی گیرند.
- سپس با نشان دادن اینکه چگونه می توانید **ویژگی های غیرخطی** را تعریف کنید، محدودیت های مدل های خطی را کاهش می دهیم، که به طور موثر پیش بینی کننده های غیرخطی را با استفاده از ماشین مدل های خطی به ما می دهد! **قالب های ویژگی** چارچوبی را برای سازماندهی مجموعه ویژگی ها در اختیار ما قرار می دهند.

مطالب مورد نظر در حوزه یادگیری ماشین

- سپس **شبکه‌های عصبی** را معرفی می‌کنیم که پیش‌بینی‌کننده‌های غیرخطی نیز ارائه می‌کنند، اما اجازه می‌دهند این غیرخطی‌ها به طور خودکار از داده‌ها یاد بگیرند. ما فوراً با **backpropagation** پیگیری می‌کنیم، الگوریتمی که به ما امکان می‌دهد بدون نیاز به گرفتن گرادیانت‌ها به صورت دستی، شیب‌های مورد نیاز برای آموزش را به صورت خودکار محاسبه کنیم.
- سپس به طور خلاصه در مورد گسترش شبکه‌های عصبی به **برنامه نویسی قابل تمایز** صحبت می‌کنیم، که به ما امکان می‌دهد به راحتی بسیاری از مدل‌های پیشرفته یادگیری عمیق موجود در NLP و بینایی کامپیوتر مانند بلوک‌های لگو را ایجاد کنیم.
- تا کنون تمرکز ما بر یادگیری تحت نظارت بوده است. ما یک مسیر انحرافی کوتاه انجام می‌دهیم و **K-means** را مورد بحث قرار می‌دهیم، که یک الگوریتم یادگیری ساده بدون نظارت برای خوشه‌بندی نقاط داده است.
- ما با یک نکته تأملی بیشتر به پایان می‌رسانیم: **تعمیم** در مورد پاسخ به این سوال است: چه زمانی یک مدل آموزش داده شده بر روی مجموعه‌ای از نمونه‌های آموزشی واقعاً به ورودی‌های تست جدید تعمیم می‌یابد؟ اینجا است که پیچیدگی مدل ظاهر می‌شود. در نهایت، **بهترین روش‌ها** را برای انجام یادگیری ماشینی در عمل مورد بحث قرار می‌دهیم.