

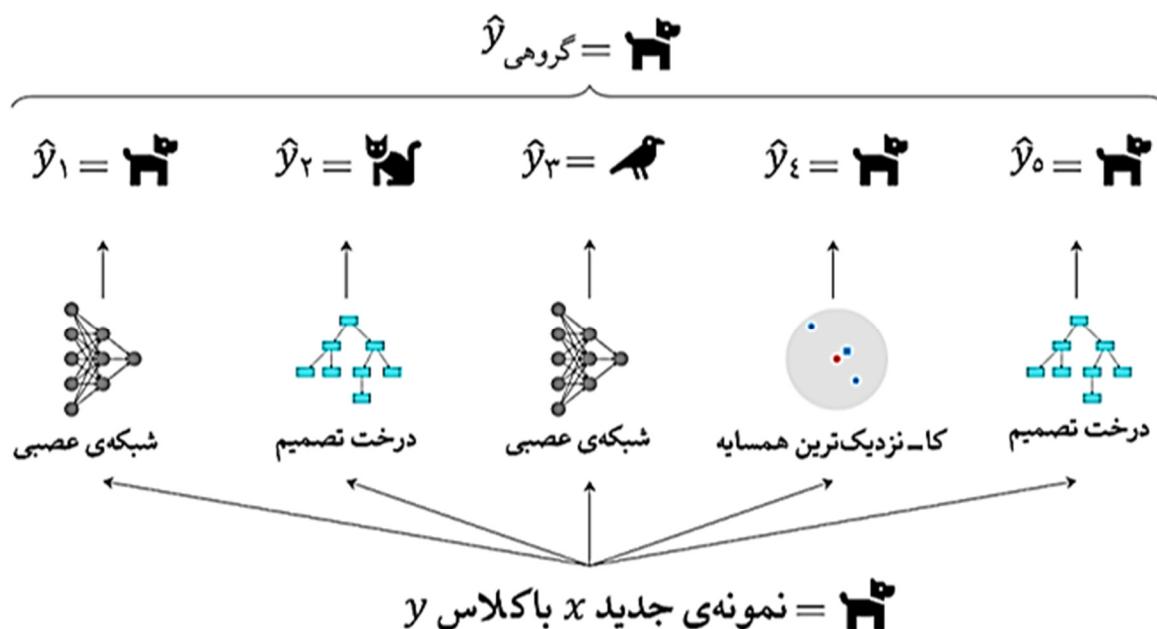
## مصطفی سبیلو - مهدی محمدی - پروژه داده کاوی

### یادگیری گروهی

در بسیاری از مواقعی که نیاز به تصمیم‌گیری مهم داریم، این طبیعت ما است که از نظر یک متخصص برای کمک به تصمیم‌گیری بهره ببریم. در بیشتر موارد، ما یک گام فراتر می‌رویم و به دنبال نظر دوم و سوم نیز هستیم. از این جهت که فکر می‌کنیم هیچ فردی به تنها بی نمی‌تواند در مورد موضوعی که از نظر عینی دشوار است (مثلًاً تشخیص پزشکی)، کاملاً داشته باشد. با در نظر گرفتن تعدادی از عوامل، ما سپس به طور شهودی، به نوعی توصیه‌های دریافت شده را برای اطلاع از تصمیم نهایی خود وزن کرده و ترکیب می‌کنیم. ما از همه این تصمیم‌ها استفاده می‌کنیم، چراکه انتظار داریم این تصمیم بهتر از تصمیمی باشد که خودمان می‌گیریم.

علیرغم مهارت ما در حل مسائل، انسان‌ها هنوز از مشورت با منابع متعدد سود می‌برند؛ بنابراین، طبیعی است که حوزه یادگیری ماشین از این عادت انسان‌ها گرفته باشد، زمانی که به دنبال راه‌هایی برای بهبود مدل‌های خود می‌گردد. حوزه‌ای از یادگیری ماشین که شامل رویه‌هایی است تا با آموزش مدل‌های متعدد و روش‌های ترکیب خروجی‌های آن‌ها به بهبود عملکرد مدل دست یابد، به عنوان یادگیری گروهی شناخته می‌شود. به عبارت ساده‌تر، یادگیری گروهی، یعنی هنر استفاده از چندین مدل، برای به دست آوردن عملکرد پیش‌بینی بهتر.

اصل اساسی پشت مدل‌های گروهی، این است که گروهی از یادگیرنده‌های ضعیف گرد هم می‌آیند تا به کمک یکدیگر یک یادگیرنده قوی تشکیل دهند. شهود پشت مدل‌سازی گروهی، مترادف با چیزی است که ما در زندگی روزمره به آن عادت کرده‌ایم؛ مانند درخواست نظر از چندین متخصص قبل از اتخاذ یک تصمیم خاص بهمنظور به حداقل رساندن احتمال تصمیم بد یا نتیجه نامطلوب. یک سناریوی نمونه که در آن یک گروه سودمند تشکیل شده است در شکل ۱ قابل مشاهده است. در این مثال، ما ۵ مدل را بر روی داده‌های آموزشی یکسان آموزش می‌دهیم هر مدل به کلاس تصویر رأی می‌دهد و گروه، کلاسی که بیشترین رأی را دارد پیش‌بینی می‌کند.



شکل ۱ - مثالی از یادگیری گروهی.

هیچ مدل یادگیری ماشین کامل نیست. همه آنها محدودیت‌هایی دارند و اشتباه می‌کنند: همان‌طور که یک انسان ممکن است قبل از تصمیم‌گیری دشوار از چندین متخصص، مشاوره بخواهد. همچنین، نشان داده شده که ترکیب پیش‌بینی‌های چندین مدل، پیش‌بینی‌های بسیار دقیق‌تری را ارائه می‌دهد. بسیاری از مطالعات نظری و تجربی موقعیت‌هایی را شناسایی کردند که در آن سیستم‌های گروهی سودمند هستند. به طورکلی، سه دلیل وجود دارد که چرا یادگیری گروهی می‌تواند نتایج بهتری نسبت به یک یادگیرنده منفرد ایجاد کند:

۱. داده‌های آموزشی محدود: برای مسائل پیچیده، داده‌ها ممکن است اطلاعات کافی را برای همه ویژگی‌های احتمالی داده‌هایی که یادگیرنده ممکن است با آنها مواجه شود، ارائه نکند؛ بنابراین، یک فرضیه منحصر به فرد برای یک مجموعه داده محدود، بعید است به دلیل اطلاعات ناکافی برای مجموعه داده دیگری کار کند. در داده‌های آموزشی با ترکیب فرضیه‌های یادگیرنده‌گان، شанс بیشتری وجود دارد که یادگیرنده گروهی با موقیت، داده‌هایی با ویژگی‌های ناشناخته را مدیریت کند.
۲. جبران فرآیندهای جستجوی ناقص: فرآیندهای جستجوی الگوریتم‌های یادگیری، ممکن است ناقص باشند. از این‌رو، حتی اگر یک فرضیه بهینه منحصر به فرد وجود داشته باشد، ممکن است هرگز پیدا نشود. یادگیری گروهی می‌تواند به جبران چنین فرآیندهای جستجوی ناقصی کمک کند.
۳. هیچ فرضیه بهینه منحصر به فردی وجود ندارد: فضای مورد جستجو برای مجموعه داده معین، ممکن است حاوی یک فرضیه بهینه واحد نباشد. از این‌رو یادگیری گروهی می‌تواند تقریب‌های خوبی را با ترکیب فرضیه‌های متعدد ارائه دهد.

### تا یادگیری عمیق هست، چرا یادگیری گروهی؟!

پیشرفت‌های محاسباتی و روش‌شناسختی اخیر، در آنچه که عموماً به عنوان یادگیری عمیق شناخته می‌شود، سودمندی درک شده از روش‌های یادگیری گروهی را تغییر داده است. با یک معماری کافی و تنظیم ابرپارامترها، شبکه‌های عصبی عمیق می‌توانند عملکردی نزدیک به کامل در طیف وسیعی از مجموعه داده‌ها داشته باشند. علاوه بر این، بسیاری از تدبیرهای منظم سازی برای اطمینان از همگرایی این مدل‌های پیچیده بدون بیش برازش در دسترس هستند.

از این‌رو، با این عملکرد فوق العاده یادگیری عمیق، آیا روش‌های گروهی هنوز نقش مفیدی دارند؟! اول اینکه، آموزش شبکه‌های عصبی عمیق، هزینه محاسباتی زیادی دارد. تنظیم ابرپارامترهای جامع برای این مدل‌ها به دلیل کمبود منابع محاسباتی به ندرت امکان‌پذیر است. با این حال، یافتن معماری مناسب برای تولید مدل‌هایی با عملکرد پیش‌بینی بالا ضروری است. برخلاف این، یک سیستم گروهی می‌تواند عملکرد پیش‌بینی مشابهی با چنین مدل‌هایی منفرد ایجاد کند، اما بدون متحمل شدن هزینه‌های محاسباتی مشابه. ممکن است در نگاه اول غیرشهودی به نظر برسد، چراکه چندین مدل برای تولید یک گروه نیز، نیاز به آموزش دارند! با این حال، مدل‌های فردی می‌توانند پیچیدگی کمتری باشند (لایه‌های کمتر) که امکان کاوش بهتر در فضای دسته‌بند را در همان بودجه محاسباتی فراهم می‌کند. علاوه بر این، بسیاری از این تنظیمات را می‌توان در سیستم نهایی گروهی (به جای انتخاب بهترین مدل) در نظر گرفت.

دوم اینکه، روش‌های منظم سازی به صورت کامل از بیش برازش جلوگیری نمی‌کنند؛ بنابراین، ترکیب مدل‌هایی که همگرا نشده‌اند، یا از مشکل بیش برازش داده‌ها رنج می‌برند، می‌توانند مدل‌های بهتری را بدون آموزش مجدد تولید کنند. درنهایت، هنگام کاوش مجموعه داده‌های دنیای واقعی، در بسیاری از مسابقات دیده می‌شود که سیستم‌های گروهی بهترین نتایج را ایجاد می‌کنند.

در حالی که شبکه‌های عصبی عمیق در برخی وظایف به عملکردی نزدیک به انسان دست‌یافته‌اند، واقعیت این است که این مدل‌ها بسیار تخصصی هستند. از این‌رو، استفاده از یک شبکه عصبی که بتواند در هنگام تشخیص بین تصاویر گربه‌ها و سگ‌ها به دقت کامل برسد و آن را در یک مسئله متفاوت اما مرتبط (مثلًاً شناسایی کلاس‌های مشابه بر اساس نمونه‌های ویدیویی) بکار

برد، عملکرد بسیار ضعیفی را در پی خواهد داشت. این به یک موضوع بسیار موربدجت معروف، به نام قضیه ناهار مجانی نیست مربوط می‌شود: هیچ مدل یادگیری ماشین بهتهایی، در همه ا نوع داده‌ها و مجموعه داده‌ها بهترین عملکرد را نخواهد داشت. این نشان می‌دهد که مجموعه‌ای از مدل‌های بسیار تخصصی، راه حل بهتری برای این موضوع هستند.

### تکنیک‌های یادگیری گروهی

یادگیری گروهی یک الگوی یادگیری ماشین است که در آن چندین مدل که اغلب "یادگیرنده ضعیف" نامیده می‌شوند برای حل یک مسئله آموزش داده می‌شوند و برای به دست آوردن نتایج بهتر ترکیب می‌شوند. فرضیه اصلی این است که وقتی مدل‌های ضعیف به درستی ترکیب شوند، می‌توانیم مدل‌های دقیق‌تری/قوی‌تری به دست آوریم.

در نظریه یادگیری گروهی، ما یادگیرنده‌گان ضعیف (مدل‌های پایه) را مدل‌هایی می‌نامیم که می‌توانند با ترکیب چندین مدل از آن‌ها به عنوان بلوک‌هایی برای طراحی مدل‌های پیچیده‌تر استفاده شوند. اغلب اوقات، این مدل‌های پایه به خودی خود چندان خوب عمل نمی‌کنند به این دلیل که بایاس یا واریانس بالایی دارند. از این‌رو، ایده روش‌های گروهی این است که سعی کنیم بایاس یا واریانس چنین یادگیرنده‌گان ضعیفی را با ترکیب چند مورد از آن‌ها، کاهش دهیم تا یک یادگیرنده قوی (مدل گروهی)، ایجاد کنیم که به عملکرد بهتری دست یابد.

برای ایجاد یک مدل یادگیری گروهی، ابتدا باید مدل‌های پایه خود را برای تجمع انتخاب کنیم. بیشتر اوقات از یک الگوریتم یادگیری پایه استفاده می‌شود تا یادگیرنده‌گان ضعیف همگی داشته باشیم که به روش‌های مختلف آموزش دیده‌اند. به این نوع مدل‌ها "همگن" گفته می‌شود.

با این حال، روش‌هایی نیز وجود دارد که از انواع مختلف الگوریتم‌های یادگیری پایه استفاده می‌کنند. سپس برخی از یادگیرنده‌گان ضعیف ناهمگن در یک "مدل مجموعه‌های ناهمگن" ترکیب می‌شوند.

یک نکته مهم این است که انتخاب یادگیرنده‌گان ضعیف ما، باید با روش تجمع این مدل‌ها هماهنگ باشد. اگر مدل‌های پایه با بایاس کم اما واریانس بالا را انتخاب کنیم، باید با روش تجمع کننده‌ایی باشد که تمایل به کاهش واریانس دارد، درحالی که اگر مدل‌های پایه با واریانس کم اما بایاس بالا را انتخاب کنیم، باید با روش تجمعی باشد که تمایل به کاهش بایاس دارد.

تکنیک‌های گروهی زیادی برای ترکیب چندین یادگیرنده ماشین در راستای ایجاد یک مدل پیش‌گویانه وجود دارد. رایج‌ترین تکنیک‌ها عبارت‌اند از: تجمعی پردازی برای کاهش واریانس و توان‌افزایی برای کاهش بایاس.

### توان‌افزایی (Boosting)

توان‌افزایی، دسته‌ای از الگوریتم‌ها است که یادگیرنده‌گان ضعیف را به یادگیرنده‌گان قوی تبدیل می‌کند. الگوریتم توان‌افزایی با آموزش یک یادگیرنده پایه شروع می‌شود و سپس توزیع نمونه‌های آموزشی را با توجه به نتیجه یادگیرنده پایه تنظیم می‌کنند تا نمونه‌های دسته‌بندی نادرست مورد توجه بیشتر یادگیرنده‌گان پایه بعدی قرار گیرد. پس از آموزش اولین یادگیرنده پایه، یادگیرنده پایه دوم با نمونه‌های آموزشی تنظیم شده آموزش می‌بیند و از نتیجه برای تنظیم مجدد توزیع نمونه آموزشی استفاده می‌شود. چنین فرآیندی تکرار می‌شود تا زمانی که تعداد یادگیرنده‌گان پایه به مقدار از پیش تعريف شده  $T$  برسد و درنهایت این یادگیرنده‌گان پایه وزن و ترکیب شوند.

شناخته‌شده‌ترین الگوریتم توان‌افزایی AdaBoost است که الگوریتم توان‌افزایی ساده را از طریق یک فرآیند تکراری بهبود می‌بخشد. ایده اصلی پشت این الگوریتم تمرکز بیشتر بر روی الگوهایی است که دسته‌بندی آن‌ها سخت‌تر است. مقدار تمرکز با وزنی که به هر الگوی مجموعه آموزشی اختصاص داده می‌شود، تعیین می‌شود. در ابتدا، وزن یکسانی به همه الگوها اختصاص داده می‌شود. در هر تکرار وزن تمام نمونه‌های دسته‌بندی اشتباه افزایش می‌یابد درحالی که وزن نمونه‌های دسته‌بندی صحیح کاهش می‌یابد. درنتیجه، یادگیرنده ضعیف مجبور می‌شود با انجام تکرارهای اضافی و ایجاد دسته‌بندهای بیشتر، بر روی نمونه‌های دشوار

مجموعه آموزشی تمرکز کند. علاوه بر این، وزنی به هر دسته‌بند اختصاص داده می‌شود. این وزن دقت کلی دسته‌بند را اندازه‌گیری می‌کند و تابعی از وزن کل الگوهای دسته‌بندی صحیح است؛ بنابراین، وزن‌های بالاتری به دسته‌بندهای دقیق‌تر داده می‌شود. از این وزن‌ها برای دسته‌بندی الگوهای جدید استفاده می‌شود.

از منظر تجزیه و تحلیل بایاس-واریانس، توان افزایی عمدتاً بر کاهش بایاس تمرکز دارد. به همین دلیل است که مجموعه‌ای از یادگیرندگان با توانایی تعیین ضعیف می‌توانند بسیار قدرمند باشند.

### تجمعی پردازی (Bagging)

تجمعی پردازی یک روش ساده و در عین حال مؤثر برای ایجاد مجموعه‌ای از دسته‌بندها است. دسته‌بند گروهی که با این روش ایجاد می‌شود، خروجی‌های دسته‌بندهای مختلف را در یک دسته‌بند واحد ادغام می‌کند. این منجر به یک دسته‌بند می‌شود که دقت آن بالاتر از دقت هر دسته‌بند منفردي است.

ایده‌پشت تجمعی پردازی، ترکیب نتایج چندین مدل برای به دست آوردن یک نتیجه کلی است. در اینجا یک سؤال وجود دارد: اگر همه مدل‌ها را بر روی یک مجموعه داده ایجاد کنیم و آن‌ها را ترکیب کنید، مفید خواهد بود؟ احتمال زیادی وجود دارد که این مدل‌ها نتیجه یکسانی داشته باشند، زیرا ورودی یکسانی دارند. پس چگونه می‌توانیم این مشکل را حل کنیم؟ یکی از این راه‌ها ایجاد یادگیرندگان پایه مختلف، با تقسیم مجموعه آموزشی اصلی به چندین زیرمجموعه غیر همپوشان و استفاده از هر زیرمجموعه برای آموزش یک یادگیرنده پایه است.

از آنجایی که زیرمجموعه‌های آموزشی متفاوت هستند، احتمالاً یادگیرندگان پایه آموزش دیده نیز متفاوت هستند. با این حال، اگر زیرمجموعه‌ها کاملاً متفاوت باشند، به این معنی است که هر زیرمجموعه فقط بخش کوچکی از مجموعه آموزشی اصلی را در بر می‌گیرد که احتمالاً منجر به یادگیری ضعیفی می‌شود. از آنجایی که یک گروه خوب مستلزم آن است که هر یادگیرنده پایه به طور معقولی خوب باشد، ما اغلب اجازه می‌دهیم زیرمجموعه‌ها به گونه‌ای همپوشانی داشته باشند که هر یک از آن‌ها شامل نمونه‌های کافی باشد.

تجمعی پردازی بر اساس نمونه‌برداری بوت استرپ کار می‌کند. با توجه به مجموعه داده با  $m$  نمونه، به طور تصادفی یک نمونه را انتخاب کرده و در مجموعه نمونه‌برداری رونوشت می‌کند. سپس، آن را در مجموعه داده‌های اصلی نگه می‌داریم به طوری که هنوز فرصتی برای برداشتن آن در دفعه بعد وجود داشته باشد. با تکرار  $m$  مرتبه این فرآیند یک مجموعه داده حاوی  $m$  نمونه به دست می‌آید که در آن برخی از نمونه‌های اصلی ممکن است بیش از یک بار ظاهر شوند درحالی که برخی ممکن است هرگز ظاهر شوند (نحویاً ۶۳٪ از نمونه‌های اصلی در مجموعه داده‌ها ظاهر می‌شوند).

اعمال فرآیند بالا در  $T$  مرتبه،  $T$  مجموعه داده را تولید می‌کند که هر کدام حاوی  $m$  نمونه است. سپس، یادگیرندگان پایه بر روی این مجموعه داده‌ها آموزش دیده و ترکیب می‌شوند. چنین رویه‌ای، روند کار اولیه تجمعی پردازی است. هنگام ترکیب پیش‌بینی‌های یادگیرندگان پایه، تجمعی پردازی روش رأی گیری ساده را برای وظایف دسته‌بندی و روش میانگین گیری ساده را برای وظایف رگرسیونی اتخاذ می‌کند. هنگامی که چندین کلاس تعداد آرای یکسانی را دریافت می‌کنند، می‌توانیم به طور تصادفی یکی را انتخاب کنیم یا اطمینان آرا را بیشتر بررسی کنیم.

نمونه‌گیری بوت استرپ مزیت دیگری را برای تجمعی پردازی به ارمغان می‌آورد: از آنجایی که هر یادگیرنده پایه فقط از ۶۳٪ از نمونه‌های آموزشی اصلی برای آموزش استفاده می‌کند، ۳۶٪ نمونه‌های باقیمانده (خارج از کیسه) را می‌توان به عنوان مجموعه اعتبارسنجی برای به دست آوردن توانایی تعیین استفاده کرد.

## زنگ تقریح

بیایید روش‌های گروهی را با استفاده از قیاس زیر تجسم کنیم: تصور کنید که باید در امتحانی شرکت کنیم که شامل ۱۰۰ سؤال درست/غلط در موضوعات مختلف از جمله ریاضی، جغرافیا، علوم، تاریخ و موسیقی است. خوشبختانه، ما اجازه داریم با پنج دوست خود - آدریانا، باب، کارلوس، دانا و امیلی - تماس بگیریم تا به ما کمک کنند. یک محدودیت کوچک وجود دارد و آن این است که همه آنها تمام وقت کار می‌کنند و وقت ندارند به همه ۱۰۰ سؤال پاسخ دهند. اما خوشحال هستند که در زیر مجموعه‌ای از آنها به ما کمک می‌کنند. از چه تکنیک‌هایی می‌توانیم برای کمک گرفتن از آنها استفاده کنیم؟ دو تکنیک ممکن به شرح زیر است:

تکنیک ۱: برای هر یک از دوستان، چندین سؤال تصادفی انتخاب کنید و از آنها بخواهید که به آنها پاسخ دهند (مطمئن شوید که هر سؤال حداقل توسط یکی از دوستان ما پاسخ داده شود). پس از دریافت پاسخ‌ها، با انتخاب گزینه‌ای که در بین کسانی که به آن سؤال پاسخ داده‌اند، بیشترین محبوبیت را داشت، به آزمون پاسخ دهید. به عنوان مثال، اگر دو نفر از دوستان ما در مورد سؤال ۱ "درست" و یکی "نادرست" پاسخ داد، ما به سؤال ۱ به عنوان "درست" پاسخ می‌دهیم (در صورت وجود مساوی، می‌توانیم یکی از پاسخ‌های برنده را به طور تصادفی انتخاب کنیم).

تکنیک ۲: امتحان را به آدریانا می‌دهیم و از او می‌خواهیم که فقط به سؤالاتی که در مورد آنها مطمئن است پاسخ دهد. ما فرض می‌کنیم که آن پاسخ‌ها خوب هستند و آنها را از آزمون حذف می‌کنیم. اکنون سؤالات باقی‌مانده را با همان دستورالعمل به باب می‌دهیم. ما به این روش ادامه می‌دهیم تا آن را به هر پنج دوست منتقل کنیم.

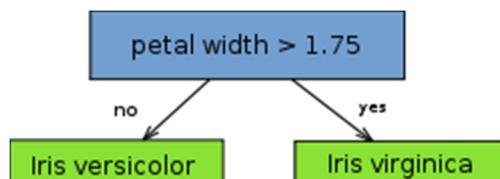
تکنیک ۱ شبیه یک الگوریتم Bagging است و تکنیک ۲ شبیه یک الگوریتم Boosting است. برای دقیق‌تر شدن، Bagging و Boosting از مجموعه‌ای از مدل‌ها به نام یادگیرنده‌گان ضعیف استفاده می‌کنند و ترکیب آنها به صورت یک یادگیرنده قوی عمل می‌کند.

### Decision Stump

استامپ تصمیم یک مدل یادگیری ماشینی است که از درخت تصمیم یک سطحی تشکیل شده است. یعنی یک درخت تصمیم با یک گره داخلی (ریشه) است که بلافاصله به گره‌های پایانی (برگ‌های آن) متصل می‌شود. یک استامپ تصمیم بر اساس مقدار تنها یک ویژگی ورودی پیش‌بینی می‌کند. گاهی به آنها ۱-قواعد نیز گفته می‌شود. معادله یک استامپ تصمیم به شکل زیر قابل بیان است:

$$h(x; j, \theta) = \begin{cases} +1 & x_j > \theta \\ -1 & \text{else} \end{cases}$$

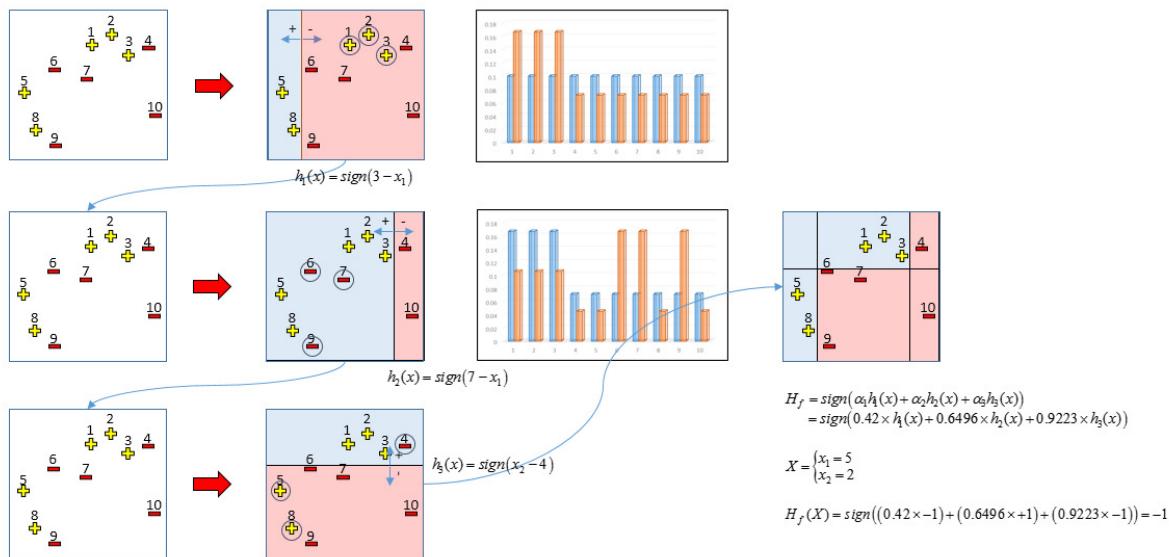
در شکل ۲ نمونه‌ای از یک استامپ تصمیم که بین دو تا از سه کلاس مجموعه داده گل زنبق تمایز قائل می‌شود به تصویر کشیده شده است.



شکل ۲ - تمایز دو کلاس از داده‌های گل زنبق با یک حد آستانه.

## مثالی شهودی:

در ابتدا احتمال انتخاب تمامی نمونه‌ها یکسان فرض می‌شود که در این مثال  $1/10$  فرض شده است. سپس با استفاده از استامپ تصمیم  $(x)$  نمونه‌های برچسب خورده را جدا می‌کنیم. مشاهده می‌شود که سه نمونه ۱، ۲ و ۳ در سمت اشتباهی قرار گرفته‌اند؛ بنابراین باید احتمال انتخاب این سه نمونه جهت استفاده در مرحله بعد افزایش پیدا کند و احتمال انتخاب سایر نمونه‌ها کاهش یابد. در مرحله بعد با استفاده از استامپ تصمیم  $(x)$  نمونه‌های برچسب خورده از جدا می‌شوند که مشاهده می‌شود سه نمونه ۷، ۶ و ۹ در سمت اشتباهی قرار گرفته‌اند؛ بنابراین احتمال انتخاب این نمونه‌ها جهت استفاده در مرحله بعد افزایش یافته و احتمال انتخاب سایر نمونه‌ها کاهش می‌یابد. همین رویه برای مرحله سوم تکرار می‌شود. در انتها با ترکیب این سه ناحیه می‌توان به مرزهای جداکننده نمونه‌های آموزشی دست یافت. حال می‌توانیم با استفاده از علامت مجموع وزن دار خطوط تفکیک کننده، برچسب نمونه جدید را مشخص کنیم. درواقع بر اساس یک رأی گیری که نسبت مشارکت هر تفکیک کننده به واسطه ضریب  $\alpha_i$  مشخص خواهد شد.



## زنگ ترجیح

برای درک این موضوع تصور کنید که ما سه دوست داریم: ترزا راست‌گو، اومبرت غیرقابل‌پیش‌بینی، و لنى دروغ‌گو. ترزا راست‌گو تقریباً همیشه حقیقت را می‌گوید، لنى دروغ‌گو تقریباً همیشه دروغ می‌گوید و اومبرت غیرقابل‌پیش‌بینی حقیقت را تقریباً نیمی از زمان می‌گوید و نیمه دیگر پنهان می‌کند. از بین این سه دوست، کدامیک کم مفید است؟ اونظوری که من می‌بینم ترزا راست‌گو خیلی قابل اعتماده چون اون تقریباً همیشه حقیقت رو می‌گه پس ما می‌تونیم بهش اعتماد کنیم در میان دو نفر دیگر، من لنى دروغ‌گو را ترجیح می‌دهم. اگر چه او تقریباً همیشه دروغ می‌گوید اما زمانی که ما از او یک سؤال بله یا خیر بپرسیم، ما به‌سادگی به عنوان حقیقت مخالف آنچه او به ما می‌گوید، و ما بیشتر اوقات درست خواهد شد! از سوی دیگر، اومبرت غیرقابل‌پیش‌بینی هیچ هدفی به ما خدمت نمی‌کند اگر ما هیچ تصوری از اینکه آیا او حقیقت را می‌گوید یا دروغ می‌گوید. در آن صورت، اگر ما به اختصاص نمره به آنچه هر دوست می‌گوید، من می‌خواهم راست‌گو ترزا نمره مثبت بالا، لنى دروغ‌گو نمره منفی بالا، و اومبرت غیرقابل‌پیش‌بینی نمره صفر است.

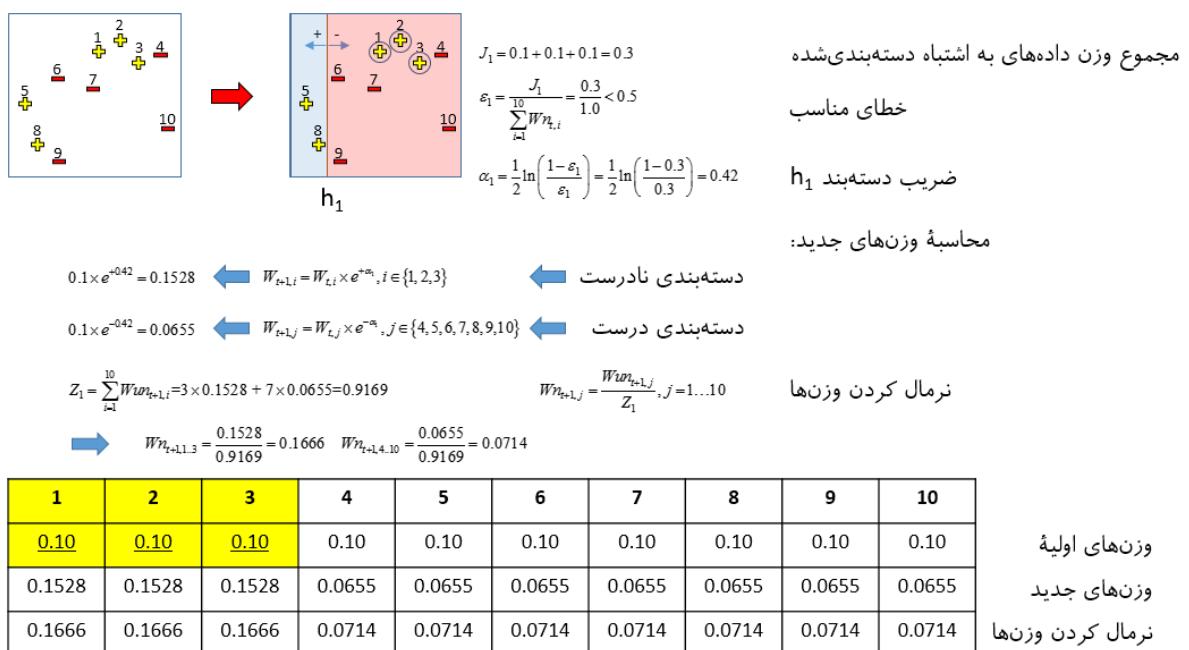
## نگاهی دقیق‌تر به محاسبات:

ابتدا مجموع وزن نمونه‌های به اشتباه دسته‌بندی شده محاسبه می‌شود ( $J_1$ ). مقدار خطای  $\epsilon_1$  با تقسیم مقدار  $J_1$  بر مجموع وزن تمامی نمونه‌ها محاسبه شده و ضریب دسته‌بند  $h_1$  توسط رابطه  $\alpha_1 = 1/2 \ln(1 - \epsilon_1 / \epsilon_1)$  به دست می‌آید. پس از آن وزن

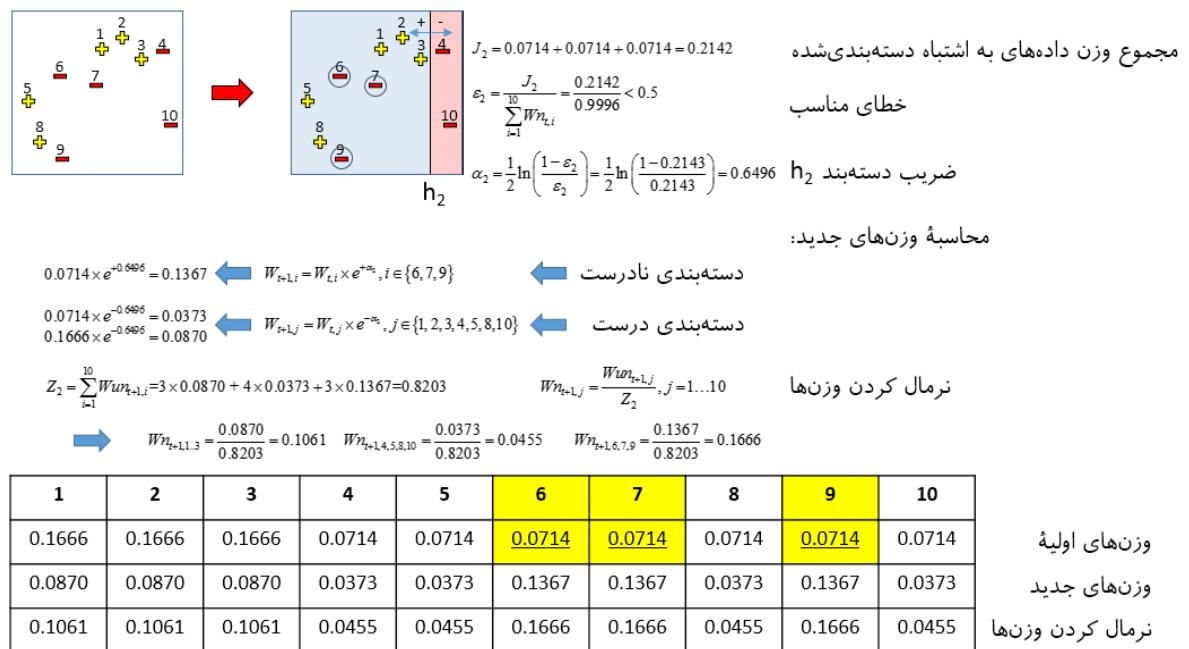
نمونه‌هایی که به صورت نادرست دسته‌بندی شده‌اند با استفاده از رابطه  $W_{t+1,i} = W_{t,i} \times e^{+\alpha_i}$  افزایش و وزن نمونه‌هایی که به صورت درست دسته‌بندی شده‌اند با استفاده از رابطه  $W_{t+1,i} = W_{t,i} \times e^{-\alpha_i}$  کاهش می‌یابند. در انتها، تمامی وزن‌های به روزرسانی شده، جهت استفاده در مرحله بعد باید نرم‌الایز شوند. مرحله دوم و سوم به نحوی مشابه با مرحله اول انجام می‌پذیرند و در انتها با استفاده از ضرایب  $\alpha_i$ -های به دست آمده از مراحل اول تا سوم، می‌توان تابع

$$H_f = sign(0.42 \times h_1(x) + 0.6496 \times h_2(x) + 0.9223 \times h_3(x))$$

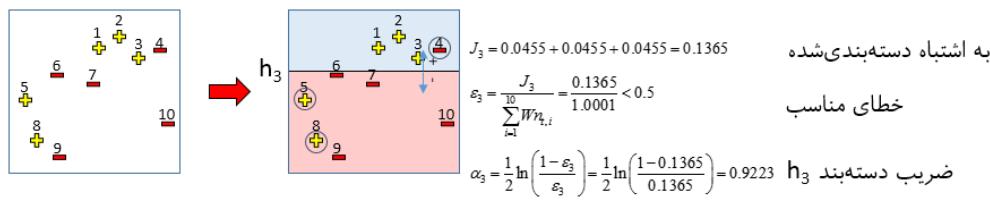
### مرحله اول:



### مرحله دوم:



### مرحله سوم:



مجموع وزن داده‌های به اشتباه دسته‌بندی شده

خطای مناسب

ضریب دسته‌بند

محاسبه وزن‌های جدید:

$$0.0455 \times e^{+0.9223} = 0.1144 \quad W_{t+1,i} = W_{t,i} \times e^{+\alpha_t}, i \in \{4, 5, 8\}$$

$$0.1061 \times e^{-0.9223} = 0.0422 \quad \text{دسته‌بندی نادرست}$$

$$0.1666 \times e^{-0.9223} = 0.0662 \quad W_{t+1,j} = W_{t,j} \times e^{-\alpha_t}, j \in \{1, 2, 3, 6, 7, 9, 10\}$$

$$0.0455 \times e^{-0.9223} = 0.0181 \quad \text{دسته‌بندی درست}$$

1	2	3	4	5	6	7	8	9	10
0.1061	0.1061	0.1061	0.0455	0.0455	0.1666	0.1666	0.0455	0.1666	0.0455
0.0422	0.0422	0.0422	0.1144	0.1144	0.0662	0.0662	0.1144	0.0662	0.0181

وزن‌های اولیه

وزن‌های جدید

### شبکه

Given:  $(x_1; y_1), \dots, (x_m; y_m)$ ,  $x_i \in X$ ,  $y_i \in Y = \{-1, 1\}$ .

Initialize  $D_1(i) = 1/m$ .

For  $t = 1 \dots T$ :

1. Train weak classifier using distribution  $D_t$ .
2. Get weak hypothesis  $h_t : X \rightarrow \{-1, 1\}$  with error  $\epsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(x_i)$
3. Choose  $\alpha_t = \frac{1}{2} \log \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)$
4. Update:

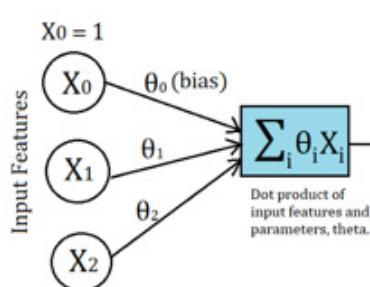
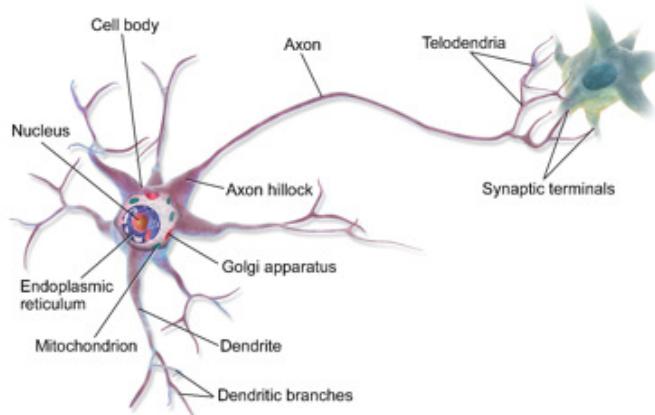
$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if instance } i \text{ is correctly classified} \\ e^{\alpha_t} & \text{if instance } i \text{ is not correctly classified} \end{cases}$$

where  $Z_t$  is a normalization factor (chosen so that  $\sum_{i=1}^m D_{t+1} = 1$ ).

Output the final hypothesis:  $H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$ .

## نگاهی گذرا به پرسپترون

شبکه عصبی پرسپترون تک لایه، یک یادگیرنده ضعیف است که می‌تواند در الگوریتم آدابوست مورد استفاده قرار گیرد. در شبکه عصبی پرسپترون مجموع وزن‌دار ورودی‌های شبکه عصبی از یکتابع فعالیت گذر کرده و بدین ترتیب برچسبی به نمونه موردنظر اختصاص می‌یابد و در صورتی که مقدار برچسب باقی متفاوت باشد تخمین صحیحی صورت نپذیرفته و بنابراین با استفاده از قانون دلتا، در یک فرایند تکراری، وزن‌ها به روزرسانی خواهند شد تا میزان خطأ به حداقل برسد.



مقدار تخمین زده شده  
مقدار واقعی

به روزرسانی وزن‌ها با استفاده از قانون دلتا

