

روش ها و الگوریتم های

تشخیص چهره^۱

^۱ Face Recognition

فهرست مطالب

4.....	قسمت اول : تشخیص چهره
5.....	فصل اول : تشخیص الگو و تشخیص چهره
7	مفاهیم اصلی تشخیص الگو و چهره
7	نگاه کلی
8	الگوها و کلاس های الگوها
9	مسائل اساسی در طراحی سیستم تشخیص الگو
11	یادگیری و تمرین دادن
12	تشخیص الگوی نظارت شده و بدون نظارت
13	کلیات یک سیستم تشخیص الگو
14	تشخیص چهره
14	کلیات یک سیستم تشخیص چهره عام
19.....	قسمت دوم : روش های استخراج ویژگی در تشخیص چهره
20.....	فصل دوم : تشخیص چهره با استفاده از چهره - ویژه ها
24	کلیات سیستم تشخیص چهره
27	محاسبه ی چهره - ویژه ها
33.....	فصل سوم : تشخیص چهره با استفاده از HMM
34	مدل مارکوف مخفی
34	HIDDEN MARKOV MODEL
34	معرفی HMM ها
35	تعریف HMM یک بعدی
36	تمرین دادن مدل و تشخیص
41	HMM ها در بینایی
44	تشریح یک معماری
46	رویه های تمرین و تست
48	توپولوژی HMM
49	ERGODIC HMMs
51	HMM های بالا - پایین
54.....	قسمت سوم : روش های دسته بندی در تشخیص چهره
55.....	فصل چهارم : تشخیص چهره با استفاده از ماشین های بردار پشتیبان
60	تشخیص چند کلاسی
62.....	فصل پنجم : تشخیص چهره با استفاده از شبکه های عصبی
63	مدل نرون
65	توابع انتقال
66	نرونی با ورودی برداری

68	معماری های شبکه
68	یک لایه از نرون ها
70	چندین لایه از نرون ها
71	یادگیری شبکه های عصبی
75	دسته بندی با استفاده از شبکه های عصبی
77.....	منابع
77.....	سایت ها

قسمت اول :

تشخیص چهره

فصل اول :

تشخيص الگو و

تشخيص چهره

چهره نقش اساسی را در شناسایی افراد و نمایش احساسات آنها در سطح جامعه دارد. توانایی انسان در تشخیص چهره ها قابل توجه است ما می توانیم هزاران چهره هی یاد داده شده در طول عمرمان را تشخیص دهیم و در یک نگاه چهره های آشنا را حتی پس از سالها جدایی شناسایی کنیم. این مهارت در مقابل تغییرات در شرایط دیداری مانند حالت چهره، سن و همچنین تغییراتی در عینک، ریش یا سبک مدل موها ایستادگی می کند. تشخیص چهره یک موضوع مهم در کاربردهایی همچون سیستم های امنیتی، کنترل کارت اعتباری و شناسایی مجرمان شده است. برای مثال، قابلیت مدل کردن یک چهره هی خاص و تمیز دادن آن از یک تعداد فراوان از مدل های چهره هی ذخیره شده، شناسایی مجرمان را به صورت گسترده ای بهبود خواهد بخشید.

اگرچه درست است که انسان ها در تشخیص چهره توانا هستند اما نحوه ی کدینگ و دی کدینگ چهره ها در مغز انسان کاملا آشکار نیست. تشخیص چهره هی انسان برای بیش از بیست سال مورد مطالعه قرار گرفته است. توسعه ی یک مدل محاسباتی برای تشخیص چهره کاملا دشوار است و دلیل آن پیچیدگی چهره ها و ساختار چند بعدی بینایی است. بنابراین تشخیص چهره یک فعالیت سطح بالا در بینایی کامپیوتر است و می تواند بسیاری از تکنیک های بینایی اولیه را در بر گیرد.

مرحله ی اول تشخیص چهره هی انسان، استخراج ویژگی های آشکار از تصاویر چهره هاست. در اینجا یک سوال بوجود می آید که تا چه اندازه ویژگی های چهره قابلیت اندازه گیری شدن را دارند. بررسی های محققین [3 و 2 و 1] در چندین سال گذشته بر آن اشاره دارد که ویژگیهای خاصی از چهره برای شناسایی چهره ها توسط انسان تشخیص داده می شود.

مفاهیم اصلی تشخیص الگو و چهره

این بخش بر روی پایه های اصلی تشخیص الگو و چهره تمرکز دارد.

نگاه کلی

نیاز به سیستم های اطلاعاتی بهبود یافته بیشتر از قبل مورد توجه قرار گرفته است چرا که اطلاعات عنصری اساسی در تصمیم سازی است و جهان در حال افزایش دادن مقدار اطلاعات در فرم های مختلف با درجه هایی از پیچیدگی است. یکی از مسائل اصلی در طراحی سیستم های اطلاعاتی مدرن ، تشخیص الگو به طور اتوماتیک است.

تشخیص به عنوان یک صفت اصلی انسان بودن است. یک الگو ، توصیفی از یک شئ است. یک انسان دارای یک سیستم اطلاعاتی سطح بالاست که یک دلیل آن داشتن قابلیت تشخیص الگوی پیشرفته است. بر طبق طبیعت الگوهای مورد تشخیص ، عملیات تشخیص در دو گونه ای اصلی تقسیم می شوند:

تشخیص آیتم های واقعی. این ممکن است به عنوان تشخیص سنسوری معرفی شود که تشخیص الگوهای سمعی و بصری را دربر می گیرد. این فرآیند تشخیص ، شناسایی و دسته بندی الگوهای فضایی و الگوهای زمانی را در بر می گیرد. مثال هایی از الگوهای فضایی کارکترها ، اثر انگشت ها ، اشیاء فیزیکی و تصاویر هستند. الگوهای زمانی شامل فرم های موجی گفتار، سری های زمانی و ... هستند.

تشخیص آیتم های انتزاعی.^۱ از طرف دیگر ، یک بحث قدیمی یا یک راه حل برای مسئله ، قابل تشخیص است. این فرآیند شامل تشخیص آیتم های انتزاعی است و می توان آن را تشخیص انتزاعی خواند.

در واقع تشخیص انسان ، یک پرسش تخمین زدن احتمال نسبی ای است که یک داده ی ورودی به یکی از مجموعه های جمعیت های آماری شناخته شده ، که وابسته به تجربیات گذشته است ، می تواند مرتبط باشد. بنابراین مسئله ی تشخیص الگو ممکن است به عنوان یک مسئله ی تشخیص اختلاف داده ی ورودی با جمعیت ها به وسیله ی جستجوی ویژگی ها یا صفات ثابت در میان اعضای یک جمعیت مورد توجه قرار گیرد.

الگوها و کلاس های الگوها

تشخیص الگو می تواند به عنوان دسته بندی داده ها ی ورودی در کلاس های شناخته شده به وسیله ی استخراج ویژگیهای مهم یا صفات داده تعریف شود.

یک کلاس الگو ، یک دسته ی متمایز شده به وسیله ی برخی صفات و ویژگی های مشترک است. ویژگی های یک کلاس الگو ، صفات نوعی هستند که بین همه ی الگوهای متعلق به آن کلاس مشترک هستند. این ویژگی ها اغلب به عنوان ویژگی های ایتراست^۲ شناخته می شوند. ویژگی هایی که تفاوت های بین کلاس های الگو را بیان می کنند اغلب به عنوان ویژگی های ایترست^۳ شناخته می شوند.

¹ Conceptual items

² intraset

³ interset

یک الگو ، توصیفی از یکی از اعضای دسته است که ارائه دهنده‌ی کلاس الگو می‌باشد. برای راحتی ، الگوها معمولاً به وسیله‌ی یک بردار نمایش داده می‌شوند.

مانند :

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ \dots \\ x_n \end{bmatrix}$$

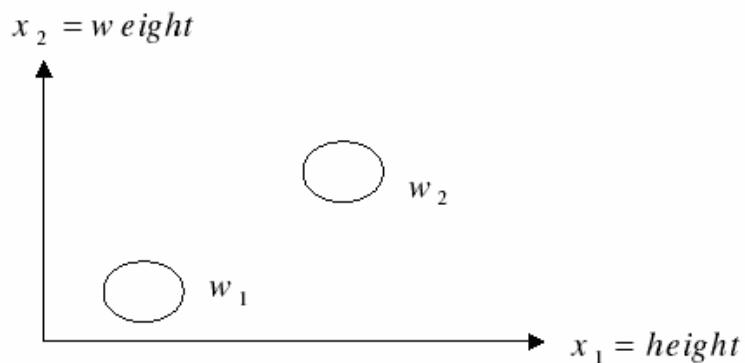
که در آن عنصر x_i یک ویژگی آن الگو را ارائه می‌کند. اغلب مفید است که یک بردار الگو به عنوان یک نقطه در فضای اقلیدسی n بعدی دیده شود.

مسائل اساسی در طراحی سیستم تشخیص الگو

به طور کلی طراحی یک سیستم تشخیص الگو چندین مسئله‌ی اصلی را در بر می‌گیرد :

- اول از همه ، ما بایستی در مورد نمایش داده‌های ورودی تصمیم بگیریم. چرا که بایستی اشیائی که قرار است مورد تشخیص قرار گیرند ، اندازه گیری شوند. این یک مسئله‌ی sensing است. هر کمیت اندازه گیری شده یک ویژگی الگو یا شیء را توصیف می‌کند و مجموعه‌ی این ویژگی‌ها در یک بردار قرار می‌گیرند. این بردار الگو که داده‌ی ورودی را توصیف می‌کند ، بایستی قالب

بندی شود. بردارهای الگو همه یا بخشی از اطلاعات اندازه گیری شده موجود را در مورد الگوها در بر می گیرد. مجموعه‌ی الگوهای متعلق به کلاسی یکسان به گروهی از نقاط در یک فضای اندازه گیری شده نگاشت می‌شوند. یک مثال ساده از این مورد در شکل 1 برای دو کلاس $w1$ و $w2$ نشان داده شده است.



شکل 1-1. دو کلاس مجزا. بردار الگو از دو کمیت $height$ و $weight$ تشکیل شده است.

- دو مین مسئله در تشخیص الگو، استخراج ویژگیها یا صفات خاصی از داده‌ی ورودی دریافته شده و کاهش ابعاد بردارهای الگوست. این مورد اغلب به عنوان مسئله‌ی پیش پردازش و استخراج ویژگی معرفی می‌شود. عناصر ویژگی‌های (ایتراست) برای همه‌ی کلاس‌های الگو مشترک هستند می‌توانند حذف شوند. اگر یک مجموعه‌ی کامل از ویژگی‌های تشخیص برای هر کلاس از داده‌های اندازه گیری شده تعیین شود. تشخیص و دسته بندی الگوها، دشواری کمتری را در برخواهد داشت. تشخیص اتوماتیک ممکن است به یک فرآیند تطبیق ساده یا یک جدول جستجو کاهش یابد. به هر حال در بسیاری از مسائل تشخیص الگو،

در عمل ، تعیین یک مجموعه کامل از ویژگیهای تشخیص اگر غیرممکن نباشد دشوار است.

● مسئله‌ی سوم در طراحی سیستم تشخیص الگو تعیین رویه‌های تصمیم بهینه است که در فرآیند شناسایی و دسته بندی مورد نیاز واقع می‌شود. پس از آنکه داده‌های مشاهده شده از الگوها جمع آوری شد و در فرم نقاط الگو یا بردارهای اندازه گیری در فضای الگو بیان شد ، ما ماشینی را می‌خواهیم تا تصمیم بگیرد که این داده به کدام کلاس الگو تعلق دارد. اجازه دهید سیستم قادر به شناسایی M کلاس مختلف الگو باشد. در این صورت فضای الگو می‌تواند شامل M ناحیه باشد که هر کدام نقاط الگوی یک کلاس را در بر می‌گیرد. مسئله‌ی تشخیص حالا می‌تواند به عنوان تولید کننده‌ی محدوده‌های تصمیم که M کلاس را روی بردارهای اندازه گیری از همدیگر جدا می‌کند در نظر گرفته شود. به طور کلی ، این محدوده‌های تصمیم به وسیله‌ی توابع تصمیم تعیین می‌شوند.

یادگیری و تمرین دادن

توابع تصمیم به روش‌های متنوعی قابل تولید هستند. زمانی که دانش قبلی در مورد الگوهایی که بایستی تشخیص داده شوند ، موجود باشد ، تابع تصمیم براساس این اطلاعات ممکن است با دقت تعیین شود. زمانی که تنها دانشی کیفی در مورد الگوها موجود باشد ، حدس‌هایی مستدل از فرم‌های تابع تصمیم می‌توان داشت. در این مورد محدوده‌های تصمیم ممکن است از پاسخ صحیح دور شود. وضعیت کلی تر آنست که دانش قبلی کمی در مورد الگوهای مورد تشخیص موجود باشد. در این شرایط ماشین‌های تشخیص الگو با استفاده از یک رویه‌ی یادگیری یا تمرین دادن طراحی بهتری خواهند داشت. به صورت ابتدائی ، توابع تصمیم موقت فرض می‌شوند و از طریق دنباله

ای از مراحل تمرینی تکراری ، این توابع تصمیم به سمت فرم های بهینه و راضی کننده پیش می روند.

این مهم است به ذهن بسیاریم که تمرین و یادگیری فقط در طول فاز طراحی سیستم تشخیص الگو انجام می شوند. هنگامی که نتایج قابل قبول با مجموعه ای الگوهای تمرینی به دست آمد ، سیستم برای وظیفه ای اجرائی واقعی خود بر روی نمونه های محیطی به کار گرفته می شود. کیفیت کارآئی تشخیص به طور گسترده ای به وسیله ای تشابه الگوهای تمرینی و داده های واقعی که سیستم در طول عملیات مواجه خواهد شد ، تعیین می شود.

تشخیص الگوی ناظارت شده^۱ و بدون ناظارت^۲.

در بسیاری موارد ، الگوهای نماینده از هر کلاس موجود هستند. در این وضعیت ها ، تکنیک های تشخیص الگوی ناظارت شده ، کاربردی هستند. پایه های این رویکرد ، مجموعه ای از الگوهای تمرینی شناخته شده برای دسته بندی و پیاده سازی یک رویه یادگیری مناسب هستند.

در برخی کاربردها ، فقط مجموعه ای از الگوهای تمرینی شناخته نشده برای دسته بندی ممکن است موجود باشند. در این موقعیت ها ، تکنیک های تشخیص چهره ای بدون ناظارت کاربرد دارند. همانطور که در بالا بیان شد ، تشخیص الگوی ناظارت یافته به وسیله ای این موضوع که دسته ای صحیح هر الگوی تمرینی مشخص است ، معرفی می شوند. در مورد بدون ناظارت ، به هر حال ، با مسئله ای یادگیری در کلاس های الگوی ارائه شده

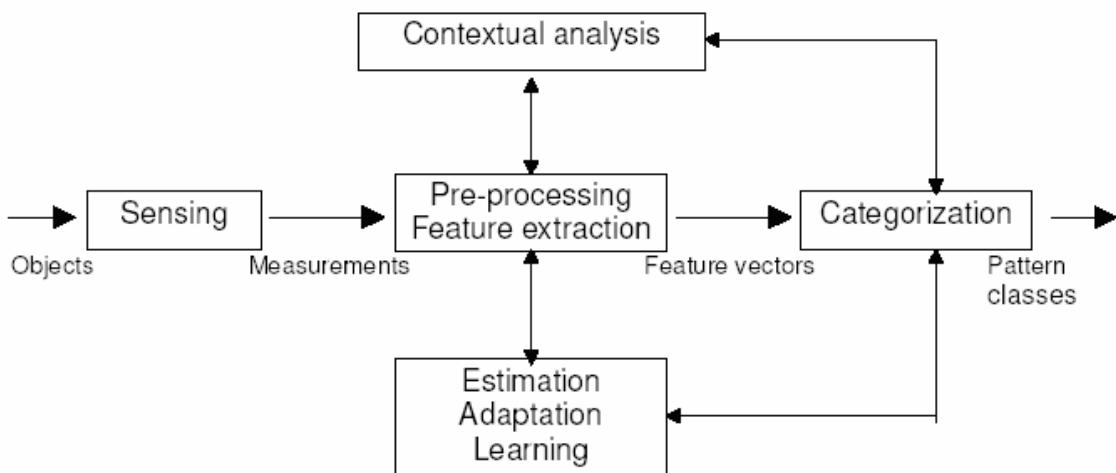
¹ Supervised

² Unsupervised

در داده ها ، مواجهیم. این مسئله همچنین به عنوان « یادگیری بدون آموزگار » شناخته می شود.

کلیات یک سیستم تشخیص الگو .

در شکل 1-2 ، دیاگرام بلوکی یک سیستم تشخیص الگوی تطبیقی نشان داده می شود.



شکل 1-2 - دیاگرام بلوکی یک سیستم تشخیص الگوی تطبیقی

تشخیص صحیح به میزان اطلاعات موجود در اندازه گیری ها و نحوه استفاده از این اطلاعات وابسته خواهد بود. در برخی کاربردها ، اطلاعات زمینه برای بدست آوردن تشخیص دقیق الزامی است. برای نمونه ، در تشخیص کارکترهای دست نویس خمیده و دسته بندي اثر انگشت ها ، اطلاعات زمینه با اهمیت هستند. زمانیکه ما قصد طراحی یک سیستم تشخیص الگو داریم که در مقابل distortion مقاوم باشد همچنین در مقابل

انعطاف پذیر باشد و نیز قابلیت خود تنظیمی را داشته باشد، ما با مسئله‌ی تطبیق دادن مواجه شده‌ایم.

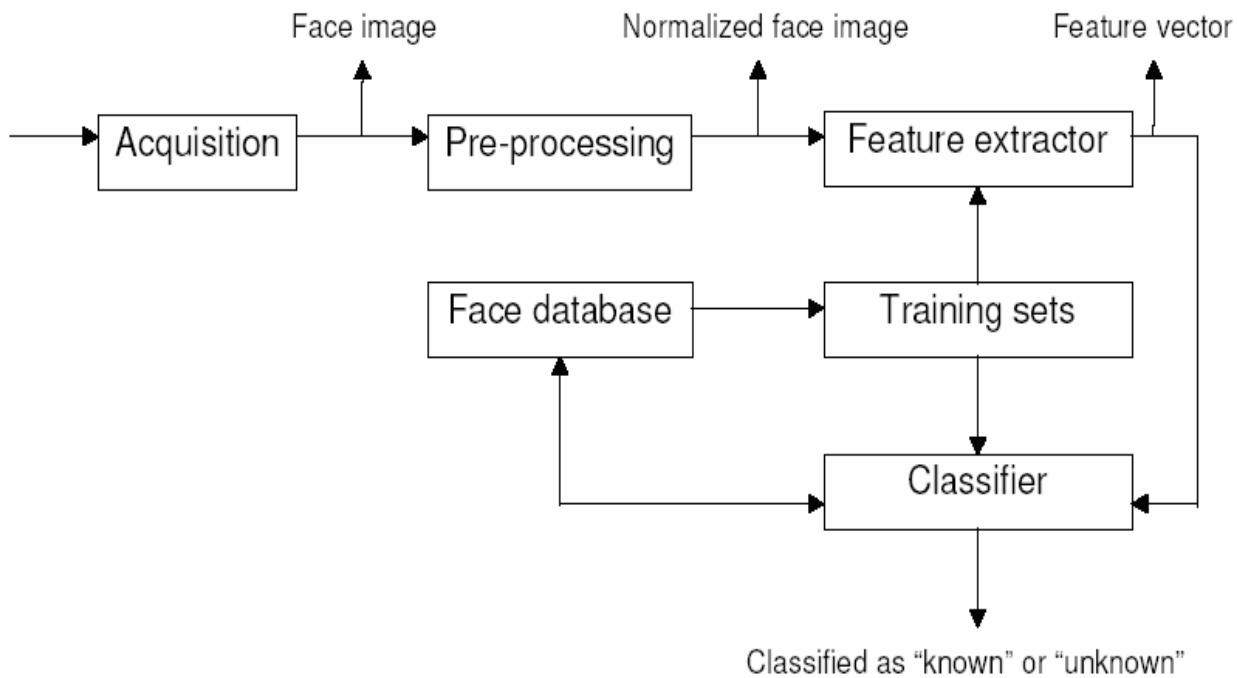
تشخیص چهره

تشخیص چهره یک عمل تشخیص الگو است که به طور خاص بر روی چهره‌ها انجام می‌شود. این عمل عبارتست از دسته‌بندی یک چهره به عنوان « شناخته شده » و « ناشناس ». پس از مقایسه با چهره‌های ذخیره شده‌ی افراد شناخته شده.

مدل‌های محاسباتی تشخیص چهره بایستی پاسخگوی چندین مسئله‌ی دشوار باشند. این سختی از آنجا ناشی می‌شود که چهره‌ها بایستی به گونه‌ای ارائه شوند که اطلاعات موجود در چهره را برای تشخیص یک چهره‌ی خاص از دیگر چهره‌ها به بهترین نحو بکار ببرند. در این مورد، چهره‌ها باعث یک مسئله‌ی دشوار می‌شوند چرا که همه‌ی چهره‌ها از آنجا که مجموعه ویژگی‌های یکسانی مانند چشم‌مان، بینی و دهان دارند، مشابه با یکدیگر هستند.

کلیات یک سیستم تشخیص چهره عام

در شکل 1-3، کلیات یک سیستم تشخیص چهره‌ی عام نشان داده شده است. این کلیات، ویژگیهایی از سیستم تشخیص الگوی عام که قبلاً بحث شد، در بر دارد.



شکل ۱-۳ - یک سیستم تشخیص چهره‌ی عام

شش بلوک عملیاتی اصلی وجود دارند که مسئولیت‌های آنها در زیر آمده است :

- **ماژول دریافت** . این نقطه‌ی ورود به فرایند تشخیص چهره است. این ماژولی است که تصویر چهره‌ی مورد نظر به سیستم ارائه می‌شود. به عبارت دیگر، در این ماژول ، از کاربر درخواست می‌شود تا یک تصویر چهره به سیستم تشخیص چهره ارائه کند. یک ماژول دریافت می‌تواند تصویر چهره را از چندین محیط متفاوت تحويل بگیرد : یک تصویر چهره ممکن است یک فایل تصویر بر روی یک دیسک مغناطیسی باشد ، ممکن است به وسیله‌ی یک frame grabber گرفته شده باشد یا به کمک یک اسکنر از روی کاغذی اسکن شده باشد.

• **ماژول پیش - پردازش.** در این ماژول ، بوسیله تکنیک های بینایی^۱ ، تصاویر چهره نرمال سازی می شوند و اگر نیاز بود ، برای افزایش کارآئی تشخیص سیستم بهبود می یابند. برخی یا تمام مراحل پیش - پردازش زیر ممکن است در یک سیستم تشخیص چهره پیاده سازی شوند :

1. نرمال سازی اندازه ای تصویر. این کار معمولاً انجام می شود تا اندازه ای تصویر گرفته شده به یک اندازه تصویر پیش فرض مانند $128*128$ تغییر کند ، این اندازه ای تصویر همان اندازه ای تصویری است که سیستم تشخیص چهره با آن کار می کند.

2. یکسان سازی هیستوگرام. این معمولاً بر روی تصاویر خیلی تاریک یا خیلی روشن برای بهبود کیفیت تصویر و بهبود کارآئی تشخیص چهره انجام می شود که گستره ای contrast تصویر را اصلاح می کند که به عنوان نتیجه ، برخی ویژگیهای مهم چهره آشکارتر می شوند.

3. فیلترینگ میانه. برای تصاویر دارای نویز ، به خصوص تصاویری که از طریق دوربین عکاسی و یا frame grabber گرفته شده است ، فیلترینگ میانه می تواند تصویر را بدون از دست رفتن اطلاعات تمیز کند.

4. فیلترینگ بالا گذر. استخراج کننده های ویژگی ای که مبتنی بر روی کلیات چهره هستند، ممکن است از تصویری که لبه یابی روی آن صورت گرفته است نتیجه ای بهتری بدهنند. فیلترینگ بالا گذر بر روی جزئیاتی مانند لبه ها تاکید دارند که در نتیجه کارآئی تشخیص لبه را افزایش می دهد.

¹ vision

5. حذف کردن پس زمینه. به منظور دست یابی به خود اطلاعات چهره ، پس زمینه ی چهره قابل حذف است. این موضوع برای سیستم های تشخیص چهره ای مهم تر است که اطلاعات موجود در کل تصویر ، استفاده می شود. همچنین توجه داشته باشیم که در حذف پس زمینه ، ماژول پیش - پردازش بایستی قادر به تعیین محدوده ی چهره باشد.

6. نرمال سازی چرخشی و انتقالی. در برخی موارد ، ممکن است بر روی تصویر چهره ای کار شود که سر شخص ، چرخش یا انتقال یافته باشد. سر، نقش کلیدی در تعیین ویژگیهای چهره بازی می کند. به خصوص برای سیستم های تشخیص چهره ای که مبتنی بر چهره های تمام رخ هستند ، ممکن است مطلوب باشد که ماژول پیش - پردازش وجود چرخش یا انتقال را مشخص کند و اگر ممکن بود ، به نرمال سازی انتقال ها و چرخش ها در مکان سر پردازد.

7. نرمال سازی نور پردازی. تصاویر چهره ی گرفته شده در شرایط نوری متفاوت می توانند کارآئی تشخیص چهره را پایین بیاورند به خصوص برای سیستم های تشخیص چهره ی مبتنی بر آنالیز جزء اصلی که اطلاعات کل تصویر برای تشخیص مورد استفاده قرار می گیرند.

• ماژول استخراج ویژگی. پس از انجام پیش - پردازش (اگر لازم بود) ، تصویر چهره ی نرمال شده به منظور پیدا کردن ویژگیهای کلیدی ای که قرار است برای دسته بندی از آنها استفاده شود ، به ماژول استخراج ویژگی فرستاده می شود. به عبارت

دیگر ، این مژول مسئول ایجاد یک بردار ویژگی است که به خوبی تصویر چهره را ارائه کند.

• **مژول دسته بندی.** در این مژول ، با کمک یک دسته بند الگو ، ویژگیهای استخراج شده از تصویر چهره با آنها یکی در کتابخانه چهره (یا پایگاه داده چهره) ذخیره شده اند مقایسه می شود. پس از انجام این مقایسه ، تصویر چهره به عنوان « شناخته شده » و « ناشناس » دسته بندی می شود.

• **مجموعه های تمرین.** مجموعه های تمرینی در طول « فاز یادگیری » فرآیند تشخیص چهره استفاده می شوند. مژول های استخراج ویژگی و دسته بندی به منظور بدست آوردن کارآئی بهینه ی تشخیص به وسیله ی استفاده از مجموعه های تمرینی ، پارامترهایشان را تنظیم می کنند.

• **پایگاه داده ی چهره.** پس از آنکه یک چهره به عنوان « ناشناخته » دسته بندی شد. تصاویر چهره می توانند با بردارهای ویژگی شان برای مقایسه های بعدی به پایگاه داده اضافه شوند. مژول دسته بندی به طور مستقیم از پایگاه داده ی چهره استفاده می کند.

ماژول های استخراج ویژگی و دسته بندی ، دو ماژولی هستند که وجه تمايز بین بیشتر سیستم های تشخیص چهره ی متفاوت را باعث می شوند. تقریبا می توان گفت که تفاوت کارائی سیستم های تشخیص چهره ، در امر تشخیص چهره ها از کارائی این دو ماژول آن ها متاثر است. در بخش های بعدی به چهار روش از روش های متفاوتی که سیستم های تشخیص چهره در این دو ماژول به کار می برند ، می پردازیم.

قسمت دوم :

روش های استخراج ویژگی

در تشخیص چهره

فصل دوم :

تشخیص چهره با استفاده از

چهره - ویژه ها

به زبان تئوری اطلاعات ، در تشخیص چهره اطلاعات داخل یک تصویر چهره ، با حداقل کارآئی استخراج و کدگذاری می شوند و سپس با یک پایگاه داده از مدل هایی که به صورت مشابه کدگذاری شده اند مقایسه می شوند. یک رویکرد ساده برای استخراج اطلاعات موجود در یک تصویر چهره ، به دست آوردن اختلاف در مجموعه ای تصاویر چهره هاست سپس می توان با استفاده از این اطلاعات به کدگذاری و مقایسه ای تصاویر چهره ای افراد پرداخت.

به زبان ریاضی ، اجزای اصلی توزیع چهره ها ، یا بردار - ویژه های ماتریس کواریانس مجموعه ای تصاویر چهره ، به گونه ای رفتار می کنند که یک تصویر به عنوان یک نقطه (یا بردار) در یک فضای با تعداد ابعاد بسیار بالا در نظر گرفته می شود.

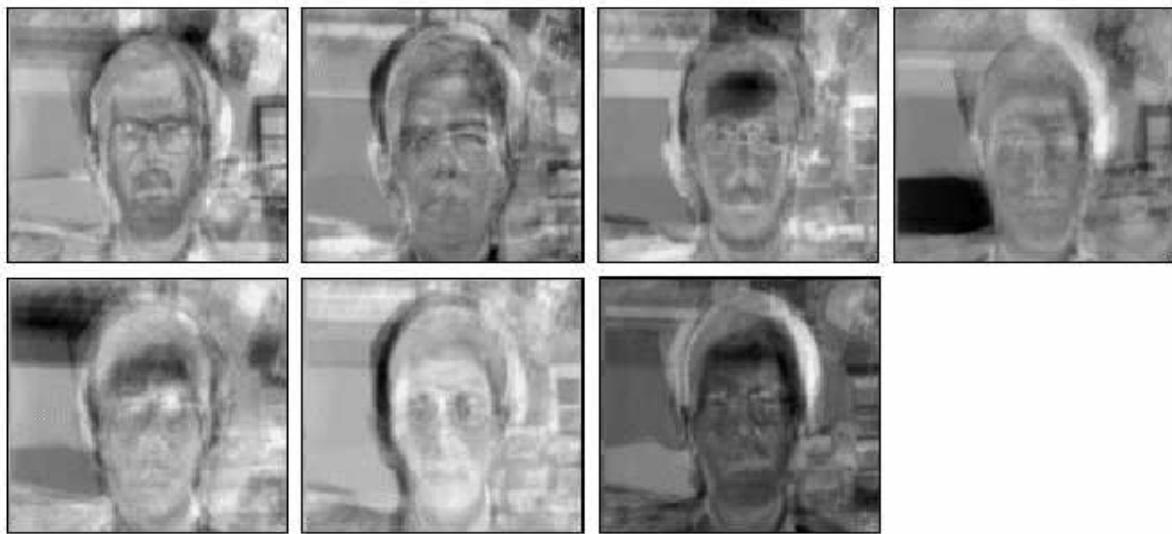
بردار ویژه ها می توانند به عنوان مجموعه ای از ویژگیها فرض شوند که باهم اختلاف بین تصاویر چهره ها را مشخص می کنند. در اینجا چون بردار ویژه ها در مورد تصاویر چهره هستند ، چهره - ویژه^۱ نامیده می شوند.

نمونه ای از تصاویر چهره در شکل 2.1.a و چهره ویژه های مربوط به آن ها در شکل 2.2 نشان داده شده است.

¹ eigenface



شکل 2.1 - (a) مجموعه‌ی تمرین تصاویر چهره به صورت نمونه (b) تصویر چهره‌ی میانگین مجموعه‌ی تمرین



شکل 2.2 - هفت چهره ویژه‌ی با بالاترین مقادیر ویژه که از مجموعه‌ی تمرین شکل 2.1 بدست آمده است.

هر چهره‌ای می‌تواند دقیقاً به وسیله‌ی ترکیب خطی چهره – ویژه‌ها ارائه شود. همچنین هر چهره می‌تواند با استفاده از «بهترین» چهره – ویژه‌ها، آنها یک بیشترین مقادیر ویژه را دارند، تقریب زده شود. $M_{چهره - ویژه}^*$ بهترین، یک زیر فضای M بعدی را به وجود می‌آورد که ما «فضای چهره»‌ی همه‌ی تصاویر ممکن می‌نامیم.

استفاده از آنالیز جزء اصلی توسعه دادند آن‌ها بهترین مختصات سیستم را برای فشرده سازی تصویر محاسبه کردند که هر محور مختصات واقعاً یک تصویر است که آن را «تصویر – ویژه» نامیدند. آن‌ها بیان کردند، حداقل در تئوری، هر مجموعه‌ای از تصاویر به صورت تقریبی می‌تواند به وسیله‌ی ذخیره کردن یک مجموعه‌ی کوچک از وزن‌ها برای هر چهره و مجموعه‌ای کوچک از تصاویر استاندارد (تصویر – ویژه‌ها) بازسازی

شود. وزن ها ای توصیف کننده ای هر چهره به وسیله ای تصویر کردن^۱ هر تصویر چهره بر روی تصویر - ویژه ها به دست می آیند.

در اینجا ما روشی که توسط M.Turk و A.Pentland [14] برای توسعه ای یک سیستم تشخیص چهره ای مبتنی بر رویکرد چهره - ویژه پیشنهاد شده را دنبال خواهیم کرد. آن ها بیان کردند که تعداد زیادی از تصاویر چهره به وسیله ای مجموع وزن داده شده ای مجموعه ای کوچکی از ویژگیهای خاص یا چهره - ویژه ها بازسازی می شوند. شاید یک روش کارا برای یادگیری و تشخیص چهره ها ، به دست آوردن ویژگیهای خاص به وسیله ای تجربه و تشخیص چهره ها به وسیله ای مقایسه ای وزن های ویژگی های مورد نیاز برای بازسازی تقریبی آنها با وزنهای مرتبط با چهره های شناخته شده باشد. بنابراین ، هر چهره به وسیله ای مجموعه ای کوچکی از ویژگی ها یا وزن های تصویر - چهره ای مورد نیاز برای توصیف و بازسازی آن ها مشخص می شود.

کلیات سیستم تشخیص چهره

سیستم تشخیص چهره ای ارائه شده در طول فرآیند تشخیص چهره از سه فاز اصلی عبور می کند. سه واحد عملیاتی اصلی در این فازها قرار گرفته اند و آنها در شکل 3.3 ترسیم شده اند. ویژگی های این فازها به همراه سه واحد عملیاتی در زیر ارائه شده اند:

- فاز شکل دهنده پایگاه داده ای چهره : در این فاز ، دریافت و پیش- پردازش تصاویر صورتی که قصد دارند به پایگاه داده ای چهره اضافه شوند ، انجام می پذیرد. تصاویر در یک پایگاه داده ای چهره در سیستم ذخیره می شوند که این پایگاه داده ای چهره را کتابخانه ای چهره می نامیم چون در آن زمان خصوصیات پایگاه داده ای رابطه

^۱ Projecting

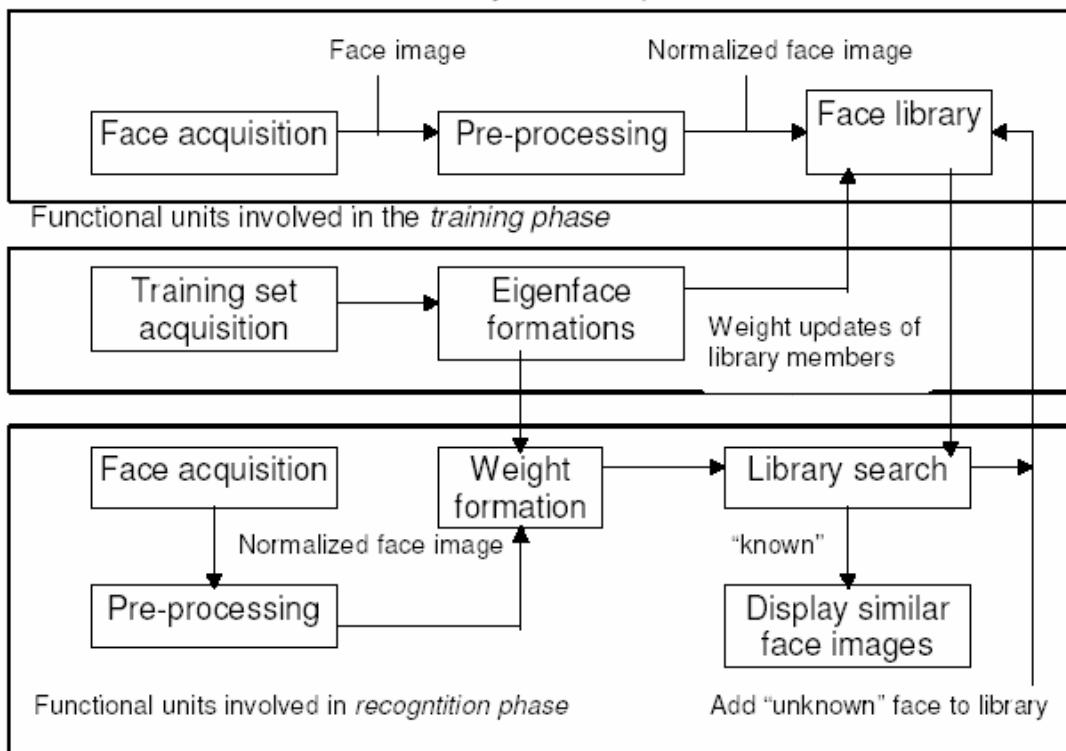
ای را ندارد. کتابخانه‌ی چهره در ابتدا خالی است. به منظور شروع فرآیندهای تشخیص چهره، در ابتدا کتابخانه‌ی خالی چهره با چهره‌ها پر می‌شود. فرض کنیم سیستم تشخیص چهره بر روی تصاویر 128×128 کار می‌کند. به منظور انجام عملیات تبدیل اندازه و بهبود تصاویر، یک ماژول «پیش-پردازش» در سیستم وجود دارد. این ماژول به صورت اتوماتیک هر تصویر چهره را به تصویر 128×128 (در صورت لزوم) تبدیل می‌کند. دیگر عملیات پیش-پردازش که قبلاً ذکر شد نیز در این ماژول قرار دارد. در کتابخانه‌ی چهره برای هر چهره نیاز به دو عنصر داریم: یک عنصر همان تصویر چهره است (که البته می‌توان فشرده سازی نیز روی آن انجام داد هر چند سرعت کم‌تر می‌شود). و دیگری بردار اوزان مربوط به هر تصویر چهره است. بردار اوزان عناصر کتابخانه‌ی چهره خالی هستند تا زمانی که یک عملیات تمرین انتخاب شود و چهره-ویژه‌ها تشکیل شوند.

- **فاز تمرین:** پس از اضافه کردن تصاویر چهره‌ها به کتابخانه‌ی چهره، سیستم برای انجام تمرین و تشکیل چهره-ویژه‌ها آماده است. تصاویر چهره‌ای که می‌خواهیم در تمرین حضور داشته باشند از کتابخانه‌ی چهره انتخاب می‌شوند. از آنجایی که عناصر کتابخانه‌ی چهره نرمال شده هستند، هیچگونه عمل پیش-پردازشی در این مرحله صورت نمی‌پذیرد. پس از انتخاب مجموعه‌ی تمرین، چهره-ویژه‌ها تشکیل و برای استفاده‌های بعد، ذخیره می‌شوند. چهره-ویژه‌ها از روی مجموعه‌ی تمرین محاسبه می‌شوند و فقط M تصویر مربوط به بیشترین مقادیر ویژه نگهداری می‌شوند. این M تصویر چهره، «فضای چهره» $\equiv M$ بعدی را نشان می‌دهند. همان طور که چهره‌های جدید تجربه می‌شوند، چهره ویژه‌ها بروز رسانی و یا دوباره محاسبه می‌شوند. اوزان مربوطه در فضای وزن M بعدی برای هر عضو کتابخانه محاسبه می‌شود (به وسیله‌ی تصویر کردن تصویر چهره‌ی آن بر روی «فضای چهره» ای که به وسیله‌ی چهره-ویژه‌ها بوجود آمده است). اکنون بردار

وزن مربوط به هر عضو کتابخانه که در ابتدا خالی بود ، به روز رسانی می شود.
سیستم حالا برای فرآیند تشخیص آماده است.

• **فاز تشخیص و یادگیری** : پس از انتخاب یک مجموعه ای تمرین و به وجود آوردن بردارهای وزن عناصر تصویر چهره ، اکنون سیستم آماده ای فرآیند تشخیص چهره است. کاربر فرآیند تشخیص را با انتخاب یک تصویر آغاز می کند. بر اساس در خواست کاربر و اندازه ای تصویر دریافتی ، مراحل پیش - پردازش برای نرمال سازی این تصویر با مشخصات کتابخانه ای چهره صورت می پذیرد. هنگامی که تصویر نرمال می شود ، بردار وزن آن به کمک چهره - ویژه ها که قبلا در فاز تمرین ساخته و ذخیره شده بود ، ساخته می شوند. پس از به دست آوردن بردار وزن ، با بردار وزن هر عضو کتابخانه ای چهره مقایسه می شود. همچنین کاربر یک آستانه شباهت را معین می کند که بر اساس آن سیستم تصاویری که کمتر از این آستانه با تصویر چهره ورودی شباهت داشته باشند به عنوان تطابق ارائه نمی کند. اگر حداقل یک تصویر چهره وجود دارد که در تشابه با تصویر ورودی مقدار آستانه را بر آورده کند ، تصویر چهره به عنوان « شناخته شده » معرفی می شود. اگر هیچ تصویری مقدار آستانه را برآورده نکند ، تصویر به عنوان « شناخته نشده » معرفی می شود. پس از آنکه تصویر به عنوان ناشناخته معرفی شد تصویر چهره ای جدید به همراه بردار وزن آن می تواند به کتابخانه ای چهره اضافه شود تا بعدا مورد استفاده قرار گیرد (یادگیری برای تشخیص).

Functional units involved in the *library formation phase*



شکل 2.3 - دیاگرام بلوکی سیستم تشخیص چهره‌ی ارائه شده

محاسبه‌ی چهره - ویژه‌ها

فرض کنید یک تصویر چهره $I(x, y)$ ، یک آرایه‌ی دو بعدی $N \times N$ از مقادیر 8-bit باشد. همچنین یک تصویر ممکن است به عنوان یک بردار N به توان 2 بعدی در نظر گرفته شود، بنابراین یک تصویر با اندازه 256×256 ، برداری 65536 بعدی یا به طور معادل یک نقطه در فضای 65536 بعدی می‌باشد.

تصاویر چهره‌ها، به خاطر شباهت در پیکربندی کلی، به طور تصادفی در این فضای بزرگ توزیع نمی‌شوند و بنابراین به وسیله‌ی یک زیر فضای به نسبت کم بعد قابل

توصیف هستند. ایده‌ی اصلی آنالیز جزء اصلی (یا بسط Karhunen - Loeve) پیدا کردن بردارهایی است که بهترین مقدار را برای توزیع تصاویر چهره در فضای تصویر کامل دارند.

این بردارها ، زیر فضایی از تصاویر چهره را تعریف می‌کنند که ما « فضای چهره » می‌نامیم. طول هر بردار N به توان 2 است ، که یک تصویر N^*N را توصیف می‌کند ، که یک ترکیب خطی از تصاویر چهره‌ی اصلی است. چون این بردارها ، بردارهای ویژه‌ی ماتریس کواریانس مربوط به تصاویر چهره‌ی اصلی هستند و چون آنها در ظاهر شبیه چهره هستند ، ما آنها را چهره – ویژه می‌نامیم.

تعاریف :

ماتریس N^*N با نام A ، بردار ویژه‌ی X و مقدار ویژه‌ی λ را دارد اگر

$$AX = \lambda X.$$

و

$$\det|A - \lambda I| = 0$$

که اگر بسط داده شود یک چند جمله‌ای درجه‌ی N با متغیر λ است که ریشه‌های آن مقادیر ویژه هستند. این اثبات می‌کند که معمولاً N مقدار ویژه وجود دارند. مقادیر ویژه‌ی مساوی از ریشه‌های مضاعف بر می‌آیند که « degenerate » نامیده می‌شود.

یک ماتریس متقارن نامیده می‌شود اگر مساوی با ترانهاده‌ی خود باشد

$$A = A^T \text{ or } a_{ij} = a_{ji}$$

آن متعامد نامیده می شود اگر ترانهاده‌ی با معکوس آن برابر باشد

$$A^T A = AA^T = I$$

و یک ماتریس حقیقی نرمال نامیده می شود اگر رابطه‌ی زیر برای آن برقرار باشد

$$AA^T = A^T A.$$

قضیه: مقادیر ویژه‌ی یک ماتریس متقارن حقیقی، همگی حقیقی هستند. در مقابل، مقادیر ویژه‌ی یک ماتریس نامتقارن حقیقی ممکن است مقادیر حقیقی را شامل شود اما همچنان ممکن است مقادیر مختلط را نیز در بر گیرد. مقادیر ویژه یک ماتریس نرمال با مقادیر ویژه‌ی «nondegenerate» کامل و متعامد هستند پس از ارائه‌ی توضیحاتی بر روی واژه‌هایی که قصد داریم در محاسبه‌ی چهره-ویژه‌ها از آن‌ها استفاده کنیم، می‌توانیم به فرآیند پیدا کردن این چهره-ویژه‌ها بپردازیم.

فرض کنید، مجموعه‌ی تصاویر چهره $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ باشند و میانگین مجموع به وسیله‌ی فرمول زیر تعریف می‌شود:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$$

بردار اختلاف هر چهره از میانگین اینگونه محاسبه می‌شود

$$\Phi_i = \Gamma_i - \Psi$$

یک مجموعه تمرین نمونه در شکل 2-1-a نشان داده شده است که چهره‌ی میانگین Ψ در شکل 2-1-b نشان داده شده است.

این مجموعه بردار‌های خیلی بزرگ موضوع کاری آنالیز جزء اصلی است که یک مجموعه از بردار‌های U_n ، orthonormal، که بهترین توصیف از توزیع داده‌ها باشند، را جستجو می‌کند. بردار K ، U_k ، انتخاب می‌شود که

$$\lambda_k = \frac{1}{M} \sum_{n=1}^M (U_k^T \Phi_n)^2$$

یک ماکریم است که

$$U_l^T U_k = \delta_{lk} = \begin{cases} 1, & \text{if } l = k \\ 0, & \text{otherwise} \end{cases}$$

بردار‌های U_k و اسکالر‌های λ_k به ترتیب بردار‌های ویژه و مقادیر ویژه‌ی ماتریس کواریانس

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T.$$

که ماتریس کواریانس $C = [\Phi_1 \ \Phi_2 \ \dots \ \Phi_M]$. ماتریس کواریانس A به هر حال ماتریس متقارن حقیقی $N^2 * N^2$ است و تعیین N^2 بردار ویژه و مقدار ویژه یک کار حجمی برای اندازه های تصویر عمومی است. ما نیاز به یک روش ممکن از نظر محاسباتی برای پیدا کردن این بردار های ویژه داریم.

اگر تعداد نقاط داده ای در فضای تصویر کمتر از بعد فضای باشد ($M < N^2$) آنچه $M-1$ به نسبت N^2 بردار ویژه ای با معنا خواهد بود. بردار ویژه های باقیمانده به مقدار ویژه ای صفر مرتبط می شوند.

به بردارهای ویژه ای V_i توجه کنید که

$$A^T A V_i = \mu_i V_i$$

با ضرب کردن هر دو طرف در A داریم

$$A A^T A V_i = \mu_i A V_i$$

می بینیم که i بردار های ویژه ای $A V_i = A A^T V_i$ هستند.

در ادامه ای این آنالیز ، ماتریس $L = A^T A$ که $M^* M$ است را می سازیم و $L_{mn} = \Phi_m^T \Phi_n$ است و M بردار ویژه ای V_i را پیدا می کنیم. این بردار ها ، ترکیب های خطی از M تصویر چهره ای مجموعه ای تمرین را برای تشکیل چهره - ویژه های U_i تعیین می کند.

$$U_l = \sum_{k=1}^M V_{lk} \Phi_k , \quad l = 1, \dots, M$$

با این آنالیز ، محاسبات از مرتبه N^2 به مرتبه M تعداد تصاویر در مجموعه M کاهش می یابند. در عمل ، مجموعه M تصاویر چهره به نسبت کوچک خواهد بود ($M < N^2$) و محاسبات کاملا قابل مدیریت می شود. مقادیر ویژه M را می دهند تا بردار های ویژه را بر اساس فایده آنها در مشخص سازی اختلاف میان تصاویر رتبه بندی کنیم.

فصل سوم :

تشخیص چهره با استفاده از

\ HMM

¹ مدل مارکوف مخفی (Hidden Markov Model)

مدل مارکف مخفی

Hidden Markov Model

روش HMM بر روی تطبیق قالب های تصویر به زنجیری از حالت های یک مدل stochastic دوبار نهفته شده بنا شده است. این فصل به پایه های اصلی HMM می پردازد و نیز چگونگی استفاده از آن را برای تشخیص چهره تشریح می کند. که این بخش ها به صورت زیر سازماندهی شده است : ابتدا یک دید کلی از HMM ها ارائه می شود. سپس برخی از کاربرد های HMM در بینایی ماشین به طور مختصر بررسی می شود و سرانجام یک معماری مبتنی بر HMM برای تشخیص چهره تشریح می شود.

معرفی HMM ها

به طور کلی HMM ها برای مدل سازی stochastic بردار سری های زمانی غیر ایستاد استفاده می شوند. آن ها یک کاربرد آشکار و فوری در پردازش سیگنال ، به خصوص تشخیص آن دارند که در این کاربرد ، سیگنال مورد نظر به طور طبیعی به صورت دنباله ای از تخمین های spectral که در طول زمان متغیر است ، ارائه می شود. Rabiner [10] یک آموزش قابل درک برای HMM ها ارائه می دهد که در بخش بعدی ملاحظه می کنیم.

تعریف HMM یک بعدی

یک HMM ، یک مدل آماری برای مجموعه ای از دنباله های مشاهده فراهم می کند. در کاربرهای گفتاری ، مشاهده ها برخی موقع فریم ها نامیده می شوند. یک دنباله مشاهده ی خاص که طول T دارد به صورت $O_1 \dots O_T$ ارائه می شود. یک HMM ، شامل دنباله ای از حالت های شماره گذاری شده 1 تا N است که به صورت یک تولید کننده ی مشاهدات ، بهتر درک می شود. حالت ها به وسیله ی کمان ها به همدیگر مرتبط می شوند و هر زمان که یک حالت j وارد می شود ، یک مشاهده بر طبق توزیع گوسی μ_j با میانگین μ_j و ماتریس کواریانس V_j مربوط به آن حالت تولید می شود. کمان ها خودشان احتمال های انتقال مربوط به خود دارند. انتقال از حالت i به حالت j ، احتمال a_{ij} دارد. احتمال شروع مدل در حالت j ، π_j است. بنابراین یک HMM به وسیله ی مجموعه ی پارامتر های زیر تعریف می شود :

N تعداد حالت ها در مدل است.

$A = \{a_{ij} : 1 \leq i, j \leq N\}$ که ماتریس انتقال حالت است.

$B = \{b_j(\cdot) : 1 \leq j \leq N\}$ کهتابع احتمال خروج از حالت است.

$\Pi = \{\pi_j : 1 \leq j \leq N\}$ که توزیع احتمال حالت اولیه است.

در نشانه گذاری مختصر شده ، مدل داده شده به صورت $\lambda = \{N, A, B, \Pi\}$ خلاصه می شود.

¹ frames

تمرین دادن مدل و تشخیص

برای یک مدل λ ارائه شده ، احتمال پشت سر هم شدن یک دنباله حالت $q_1 \dots q_T$... $Q = q_1$ و دنباله ای مشاهدات مربوطه $O = o_1 \dots o_T$ به وسیله ای ضرب هر احتمال انتقال در هر احتمال خروج در هر مرحله ای t به صورت زیر محاسبه می شود.

$$P(\mathbf{O}, Q | \lambda) = \pi_{q_1} b_{q_1}(\mathbf{o}_1) \left[\prod_{t=2}^T a_{q_{t-1}, q_t} b_{q_t}(\mathbf{o}_t) \right] \quad (3.1)$$

در عمل ، دنباله حالت ناشناخته است و معادله ای 3.1 قابل محاسبه نیست. به هر حال احتمال $(\mathbf{O} | \lambda)$ به وسیله ای جمع کردن همه ای دنباله حالت های ممکن قابل محاسبه است.

$$P(\mathbf{O} | \lambda) = \sum_Q P(\mathbf{O}, Q | \lambda) \quad (3.2)$$

جادبه ای کلیدی HMM ها آن است که یک رویه ای ساده برای پیدا کردن پارامتر های λ که معادله ای 3.2 را ماکریم می کنند ، وجود دارد. این رویه معمولاً به عنوان تخمین 'Mjugad' Baum-Welch¹ معروفی شده به [11] ارجاع می شود و برای عملیاتش به الگوریتم رو به جلو - رو به عقب نیاز دارد که در آن احتمال رو به جلو² $P(o_{t+1} \dots o_T | q_t = j, \lambda)$ و احتمال رو به عقب $(\mathbf{O}_t, q_t = j | \lambda)$ به طور کارایی به وسیله ای یک تکرار ساده تعریف می شود. متغیر های رو به جلو و رو به عقب $(\alpha_t(j)$ و $(\beta_t(j)$ به صورت زیر تعریف می شوند :

¹ re-estimation

² forward - backward

$$\alpha_t(j) = P(\mathbf{o}_1 \dots \mathbf{o}_t, q_t = j | \lambda) \quad (3.3)$$

$$\beta_t(j) = P(\mathbf{o}_{t+1} \dots \mathbf{o}_T | q_t = j, \lambda) \quad (3.4)$$

سپس متغیر ها می توانند پیدا شوند :

1. مقدار دهی اولیه

$$\alpha_1(j) = \pi_j b_j(\mathbf{o}_1), \quad 1 \leq j \leq N \quad (3.5)$$

$$\beta_T(j) = 1, \quad 1 \leq j \leq N \quad (3.6)$$

2. در ادامه

$$\begin{aligned} \alpha_{t+1}(j) &= \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(\mathbf{o}_{t+1}), & 1 \leq t \leq T-1 \\ && 1 \leq j \leq N \end{aligned} \quad (3.7)$$

$$\begin{aligned} \beta_t(i) &= \sum_{j=1}^N a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j), & t = T-1, \dots, 1 \\ && 1 \leq i \leq N \end{aligned} \quad (3.8)$$

3. در پایان

$$P(\mathbf{O} | \lambda) = \sum_{i=1}^N \alpha_T(i) = \sum_{i=1}^N \beta_1(i) \quad (3.9)$$

TN^T محاسبه‌ی مستقیم $(\mathbf{O} | \lambda)$ بر طبق معادله‌ی 3.2 از مرتبه‌ی محاسباتی است. در حالیکه محاسبات از طریق الگوریتم رو به جلو - رو به عقب معادله‌ی 3.9 از مرتبه‌ی محاسباتی TN^2 است که در [10] Rabiner تشریح شده است.

ضرب احتمال های رو به جلو و رو به عقب در حالت نرمال شده ، احتمال قرار گرفتن در حالت j در مرحله t دنباله χ مشاهده O را به دست می دهد. این متغیر به صورت $\gamma_t(j) = P(q_t=j|O, \chi)$ تعریف می شود و به صورت زیر محاسبه می شود :

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)} \quad (3.10)$$

با این احتمال قرارگیری حالت ، یک مجموعه χ جدید از پارامتر های HMM برای هر حالت j ، به وسیله χ محاسبه χ میانگین های وزن داده شده ، پیدا می شود. مدل توصیف شده به وسیله χ مجموعه χ جدید پارامتر ها به صورت $\chi = \{\bar{\Pi}, \bar{A}, \bar{B}\}$ تعریف می شود. برای تخمین پارامتر های انتقال ، کمیت مرتبط $\xi_t(i, j)$ به صورت احتمال بودن در حالت i در زمان t و در حالت j در زمان $t+1$ در مدل و دنباله χ مشاهده O ارائه شده ، تعریف می شود.

$$\xi_t(i, j) = P(q_t=i, q_{t+1}=j|O, \lambda) \quad (3.11)$$

با استفاده از تعاریف احتمال های رو به جلو و رو به عقب ، این می تواند به صورت زیر بیان شود :

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \quad (3.12)$$

با استفاده از مفهوم رخداد های شمارش^۱ ، پارامتر های مدل برای $\bar{\chi}$ می تواند به صورت زیر تخمین مجدد زده شود :

$$\bar{\pi}_i = \gamma_1(i) \quad (3.13)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (3.14)$$

$$\bar{\mu}_i = \frac{\sum_{t=1}^T \gamma_t(i) \cdot \mathbf{o}_t}{\sum_{t=1}^T \gamma_t(i)} \quad (3.15)$$

$$\bar{V}_i = \frac{\sum_{t=1}^T \gamma_t(i) \cdot (\mathbf{o}_t - \bar{\mu}_i)(\mathbf{o}_t - \bar{\mu}_i)'}{\sum_{t=1}^T \gamma_t(i)} \quad (3.16)$$

که علامت پریم بر ترانهاده i بردار اشاره دارد و $\bar{\mu}_i$ و \bar{V}_i تخمین های میانگین و ماتریس کواریانستابع احتمال خروجی گوسی برای حالت i هستند. همه ی چیز های بالا در مورد یک دنباله ی مشاهده ی منفرد توصیف شده اند. اما برای بسط این به ماکریمم ساختن مجموعه ای از دنباله های مشاهده ، به وسیله ی جمع کردن denominators و numerators در فرمول تخمین مجدد بر همه ی دنباله ها ، جزئی است. بنابراین ، با استفاده از یک یا چند دنباله ی مشاهده ی تمرینی شناخته شده ، که از یک کلاس خاص هستند ، پارامتر های یک HMM می توانند برای شکل دادن یک مدل آماری برای ارائه ی آن کلاس تخمین زده شوند و یک مدل متفاوت برای هر کلاس شکل می گیرد.

به منظور استفاده از HMM ها برای تشخیص ، دنباله ی مشاهده از سیگنال تست بدست می آید و سپس احتمال اینکه این سیگنال توسط هر یک از HMM ها تولید شود، محاسبه می شود. HMM ای که بیشترین احتمال را دارد سیگنال تست را شناسایی می

^۱ counting occurrences

کند. این احتمال بایستی دقیقاً احتمال کل تعریف شده در معادله ۳.۲ باشد. به هر حال، در عمل آسان‌تر آن است که دنباله‌ی حالتی که معادله ۳.۱ را ماقزیم می‌کند پیدا کنیم و در عوض احتمال ماقزیم مربوطه را استفاده کنیم که با یک الگوریتم برنامه نویسی پویا قابل محاسبه است. بهترین امتیاز در طول یک مسیر منفرد (به صورت یک دنباله از حالات) در زمان t ، که برای اولین مشاهده‌ی t محاسبه می‌کند و در حالت j پایان می‌پذیرد، به وسیله‌ی کمیت $\delta_t(j)$ به صورت زیر تعریف می‌شود:

$$\delta_t(j) = \max_{q_1, \dots, q_{t-1}} \{P(q_1 \dots q_t = j, \mathbf{o}_1 \dots \mathbf{o}_t | \lambda)\} \quad (3.17)$$

یک رویه‌ی ابتدایی که برای محاسبه‌ی مقادیر $\delta_t(j)$ قابل استفاده است به صورت زیر است:

مقدار دهی اولیه

$$\begin{aligned} \delta_1(j) &= P(q_1 = j, \mathbf{o}_1 | \lambda) \\ &= P(q_1 = j | \lambda)P(\mathbf{o}_1 | q_1 = j, \lambda) \\ &= \pi_j b_j(\mathbf{o}_1), \end{aligned} \quad (3.18)$$

$$1 \leq j \leq N$$

در ادامه

$$\begin{aligned} \delta_{t+1}(j) &= \max_{1 \leq i \leq N} [\delta_t(i)a_{ij}]b_j(\mathbf{o}_{t+1}), \\ 1 \leq t &\leq T-1, \\ 1 \leq j &\leq N \end{aligned} \quad (3.19)$$

در پایان

$$P^* = \max_{1 \leq j \leq N} [\delta_T(j)] \quad (3.20)$$

کمیت P^0 احتمال پشت سر هم بودن دنباله حالت بهینه و دنباله مشاهده O از ارائه شده به مدل λ ، است. این مقدار احتمال ماکزیمم برای تشخیص استفاده می شود. یک آرایه $\Psi_t(j)$ برای keep track کردن حالت هایی که معادله $i = 3.17$ را ماکزیمم می کنند، تعریف می شوند. دنباله حالت احتمال ماکزیمم q_1, q_T, \dots, q_T می تواند به وسیله $\Psi_t(j)$ عقب روی از طریق $\Psi_t(j)$ به صورت زیر پیدا شود.

$$q_T = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

For $t = T - 1$ to 1

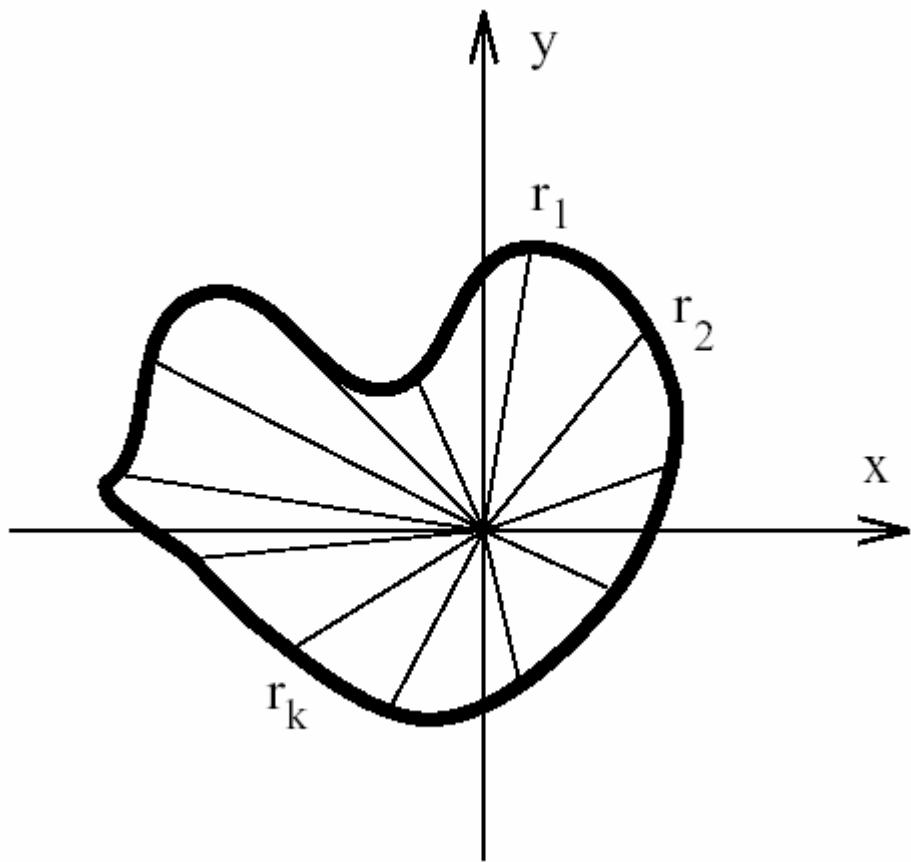
$$q_t = \psi_{t+1}(q_{t+1})$$

HMM ها در بینایی

در بیشتر موارد HMM ها در کاربردهای تشخیص گفتار مورد استفاده قرار گرفته اند که در این زمینه به طور عمیق مورد مطالعه قرار گرفته اند و یک تکنیک جا افتاده در این زمینه به حساب می آیند. همانطور که قبل گفته شد، HMM ها ویژگی های آماری دنباله های مشاهده O را مدل می کنند و داده i گفتاری به طور طبیعی بر روی محور زمان یک بعدی است.

HMM های با چگالی پیوسته را با مدل داده autoregressive برای دسته بندی اشکال دو بعدی بسته ، ترکیب کردند. هر شکل با استفاده از یک دنباله ی شعاعی از مرکز ثقل^۱ شکل برای محیط شکل نشان داده شده در تصویر 3.1 ارائه شد. هر شعاع با استفاده از یک ترکیب خطی از M شعاع قبلی به علاوه یک مقدار ثابت و یک مقدار error پیش بینی می شد. دنباله در T سگمنت L عنصری تقسیم شده بود. هر بردار ویژگی شعاعی شامل m ضریب autoregressive برای شعاع کنونی ، نسبت ثابت به مقدار error و میانگین سگمنت کنونی می شود. آن ها با هشت کلاس از اشیاء تجربه ای داشتند و یک HMM مشخص را برای هر کلاس با استفاده از 20 نمونه تمرین دادند. برای تست کردن از نمونه های بیشتر از 10 کلاس استفاده شد که دقت تشخیص شیء تا ۱۰۰٪ گزارش شد.

^۱ gravity



شکل ۱-۳. تکنیک نمونه برداری شعاعی

کاربرد دیگر HMM های گستته به وسیله‌ی kundu[13] برای کاربرد های تشخیص دست نوشته مورد مطالعه قرار گرفت. یک HMM برای مدل کردن حروف الفبا ساخته شد. هر حرف با یکی از حالت های HMM گستته معرفی شد.

مقاله های گوناگون روی کاربرد های HMM ها برای تشخیص متن و کاراکتر نوری Agazzi ، Agazzi[15]، [14] Pieraccini و Levin [16] کاری را روی تشخیص متن با استفاده از HMM های planar بهبود یافته ارائه دادند.

تشریح یک معماری

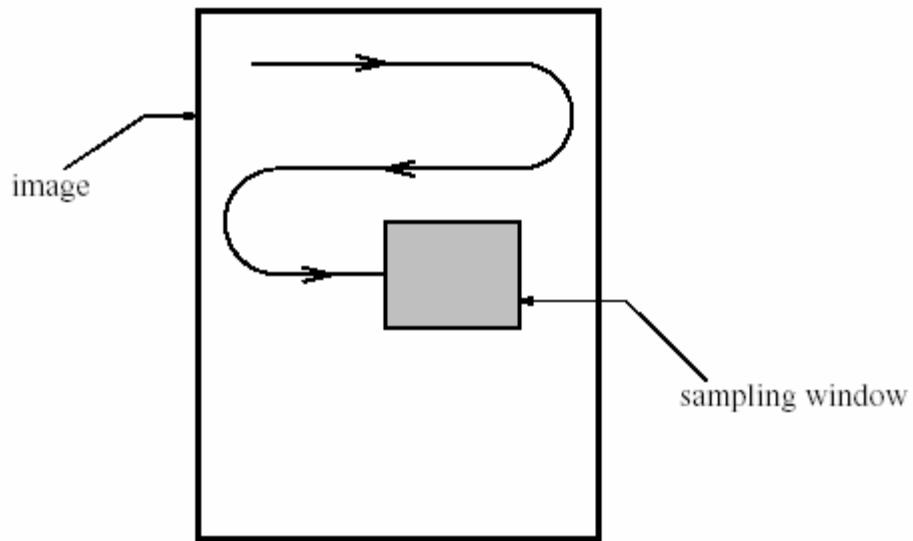
در ادامه ، مسئله‌ی تشخیص چهره از پرسپکتیو تشخیص الگوی آماری بررسی می‌شود. به طور معمول ، یک چهره قابل تقسیم به تعدادی ناحیه مانند دهان ، چشم ، بینی و ... است و اگر مکان این نواحی به طور مطمئنی قابل بدست آوردن بود ، آنگاه تکنیک‌های استاندارد مطابقت الگو ، به طور مجزا برای هر ناحیه به جهت محاسبه‌ی یک متريک فاصله‌ی کلی ، قابل استفاده هستند. به هر حال ، در عمل مکان یابی نواحی کار بسیار دشواری است.

یک راه حل ممکن برای مسئله‌ای که در بالا ذکر شد ، آن است که بین نواحی چهره و حالت‌های HMM چگالی پیوسته ، ارتباط برقرار کنیم. در پی اين کار ، اين امكان به وجود می‌آيد تا محدوده‌های بین نواحی به وسیله‌ی انتقال‌های احتمالی بین حالت‌ها ارائه شوند و تصویر واقعی در یک ناحیه به وسیله‌ی یک توزیع گوسی multivariate مدل شود. در کل ، HMM نیاز به دو بعدی بودن دارد. به هر حال ، در ادامه ، به یک تقریب مرتبه اول^۱ در شرایطی که نواحی چهره ، محدود به باند‌های افقی هستند ، دست می‌زنیم.

در اينجا ، يک مدل با استفاده از چندين تصویر چهره برای يک موضوع يكسان تمرین داده می‌شود. هر تصویر يک دنباله‌ی مشاهده $O = o_1 \dots o_T$ تولید می‌کند. مشاهده‌ی o_t از يک بلاک شامل پيكسل‌ها در تصویر دو بعدی به وسیله‌ی ميانگين‌های يک پنجره نمونه برداری که تصویر را به ترتیبی که در شکل 3.2 نشان داده اسکن می‌کند ، به

^۱ first – order approximation

دست می آید. پیکسل های پنجره‌ی نمونه برداری شامل مقادیر سطح شدت، در بردار ستونی t مرتب می شوند. فرآیند نمونه برداری مقداری که یک مدل می تواند موفق باشد را تعیین می کند. HMM‌ها توزیع آماری همه‌ی دنباله‌های مشاهده‌ی ارائه شده می مرتب با یک کلاس خاص یا در این زمینه مرتب با یک موضوع خاص را ارائه می کنند. در مورد تصاویر چهره، تعدادی از دید‌ها و حالات^۱ مختلف هر چهره قابل ترکیب در یک مدل آماری منفرد است.



شکل 2-3. تکنیک نمونه برداری برای 1D HMM

در ادامه دو توپولوژی پایه‌ای HMM بررسی می شوند: ergodic و بالا – پایین^۲.

توپولوژی ergodic مدلی ساده و بدون قید از داده‌ها تولید می کند. این مدل هیچ استفاده‌ای از اطلاعات ساختاری نمی کند.

¹ expressions

² top - bottom

رویه های تمرین و تست

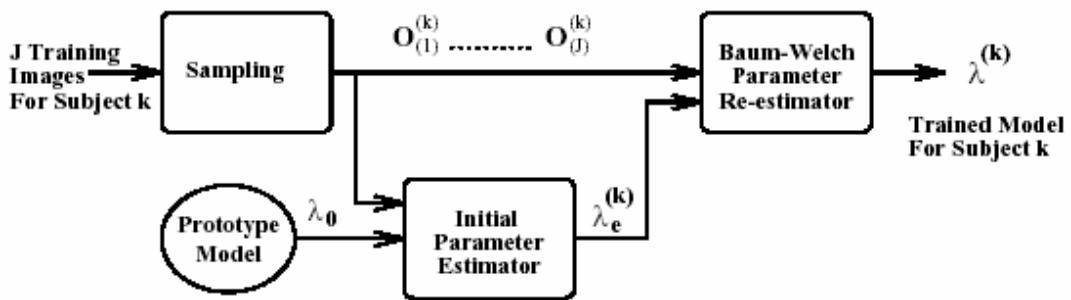
فرآیند تمرین دادن برای هر موضوع S در پایگاه داده شامل مراحل زیر است که در دیاگرام شکل 3-3 نیز خلاصه شده است :

1. تعداد J تصویر تمرین برای K امین موضوع در پایگاه داده جمع آوری می شود و برای تولید J دنباله‌ی مشاهده‌ی مشخص نمونه برداری می شوند.

2. مدل HMM نمونه λ_0 با هدف تعیین تعداد حالت‌های HMM ، انتقال‌های مجاز بین حالت‌ها و اندازه‌ی بردار‌های دنباله‌ی مشاهده ساخته می شود.

3. مجموعه‌ای از مقادیر پارامتر‌های اولیه با استفاده از داده‌های تمرینی و مدل نمونه به روش تکرار محاسبه می شود. در دور اول ، داده به طور یکنواخت بخش بندی می شود و هر بخش به یک حالت مدل منطبق می شود. در دورهای بعدی ، بخش بندی یکنواخت با حاشیه بندی Viterbi جایگزین می شود. نتیجه یک HMM اولیه تخمین $\chi_e^{(k)}$ که به عنوان ورودی به مرحله‌ی تخمین مجدد استفاده می شود.

4. پارامتر‌های HMM با استفاده از روش Baum-Welch مجدد تخمین زده می شوند. پارامتر‌های مدل برای ماکریمم سازی محلی احتمال مشاهده‌ی داده‌ی تمرینی برای هر مدل تنظیم می شوند. نتیجه‌ی این فرآیند یک HMM به نام $\chi^{(k)}$ است که برای ارائه‌ی موضوع k در پایگاه داده استفاده می شود.



شکل 3-3. دیاگرام بلوکی تکنیک تمرین دادن

تشخیص به وسیله‌ی یک تشخیص دهنده‌ی Viterbi انجام می‌شود. تصویر تست با هر یک از HMM‌های ارائه دهنده‌ی موضوعات مختلف مطابقت داده می‌شود و بیشترین مطابقت انتخاب می‌شود. روند تشخیص مراحل زیر را شامل می‌شود که در دیاگرام شکل 3-4 خلاصه شده است:

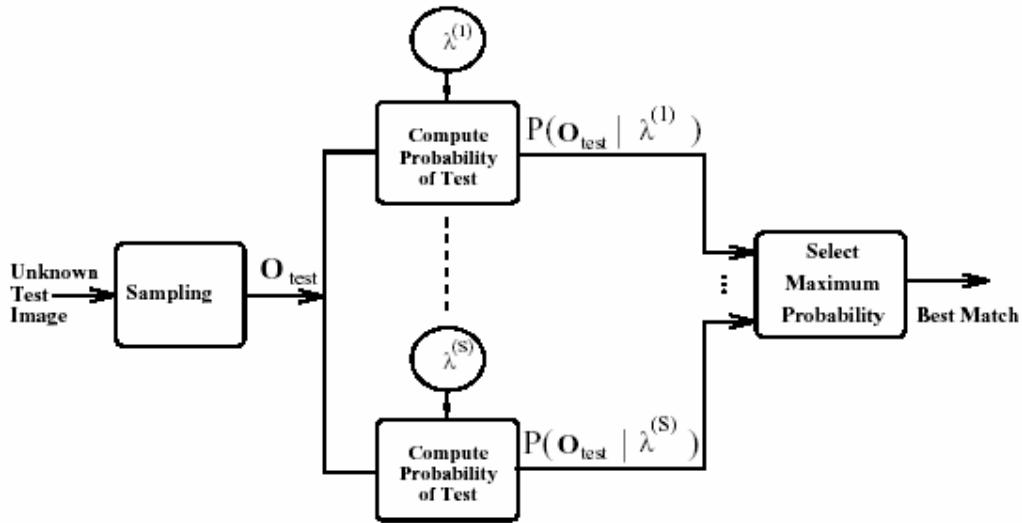
1. تصویر تست نا شناخته برای تولید دنباله‌ی مشاهده O_{test} نمونه برداری می‌شود.

2. دنباله‌ی مشاهده با هر مدل چهره به وسیله‌ی محاسبه‌ی احتمال مدل مطابقت داده می‌شود:

$$P(O_{test} | \lambda^{(k)}), \quad 1 \leq k \leq S$$

در عمل همانطوری که در بخش قبلاً شرح داده شد، مقدار احتمال ماکزیمم استفاده می‌شود.

3. مدلی با بیشترین احتمال انتخاب می‌شود و این مدل هویت چهره‌ی ناشناس را معرفی می‌کند.



شکل 4-3. دیاگرام بلوکی تشخیص دهنده ی چهره

HMM توپولوژی

یک HMM دو بعدی کاملاً مرتبط، برای مدل سازی یک تصویر دو بعدی مطلوب است. به هر حال پیچیدگی محاسباتی برای یک شبکه ی دو بعدی کاملاً مرتبط، همانطور که در [14] بحث شده، نمایی است. روش های ارائه شده در ادامه چگونگی تبدیل تصویر دو بعدی در دنباله های یک بعدی قابل استفاده در آنالیز HMM یک بعدی را دنبال می کنند.

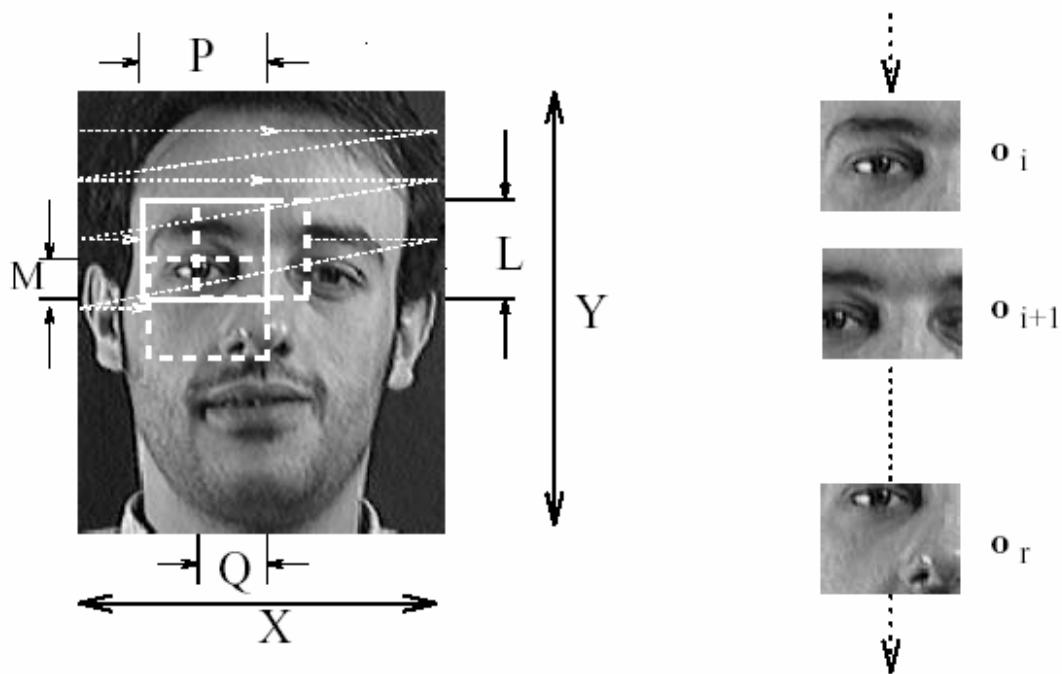
تکنیک های نمونه برداری که در ادامه معرفی می شوند، هدف استخراج یک دنباله ی مشاهده ی یک بعدی از تصویر دو بعدی را دنبال می کنند. دنباله ی مشاهده به وسیله ی استخراج بلوک های پیکسل از یک تصویر با استفاده از پنجره ی نمونه برداری ساخته می شود. هر مشاهده یک بردار ستونی شامل سطح شدت پیکسل ها در پنجره است. دنباله به

وسیله‌ی اسکن کردن با یک ترتیب تشکیل می‌شود. روشی که پنجره‌ی نمونه برداری تصویر را اسکن می‌کند بر توپولوژی مدل که داده‌ها را توصیف می‌کند، تاثیر گذار است. در بخش‌های زیر، دو تکنیک نمونه برداری مختلف آنالیز می‌شوند. در مورد اول، پنجره‌ی نمونه برداری مستطیلی تصویر را از چپ به راست و از بالا به پایین اسکن می‌کند. که یک مدل ergodic برای ارائه‌ی دنباله‌ها استفاده می‌شود. در مورد دوم، پنجره‌ی نمونه برداری بلوك‌هایی از خطها را از بالا به پایین چهره استخراج می‌کند. با استفاده از این مزیت موجود که ویژگی‌ها (چشم‌ها، بینی، دهان و ...) با بک ترتیب قابل پیش‌بینی ظاهر می‌شوند، یک HMM غیر ergodic (بالا - پایین) برای ارائه‌ی تصویر چهره ساخته می‌شود. مدل ergodic از اطلاعات ساختاری هیچ استفاده‌ای نمی‌کند. از طرف دیگر رویکرد بالا - پایین، از الگوهای داده‌ای تا حدودی بهره می‌برد.

Ergodic HMMs

در مدل‌های ergodic، هر حالت، از هر حالت دیگر قابل دسترسی است. این بدان معناست که همه‌ی ضرایب ماتریس انتقال A مثبت هستند. به طور کلی مدل‌های ergodic زمانی که قید‌های محدود برای سیگنال می‌توانند بکار روند، استفاده می‌شوند و کلی ترین نوع HMM‌ها هستند. برای تشریح نحوه‌ی کار آن‌ها، یک ergodic HMM برای یک تصویر نمونه برداری شده با استفاده از شکل 3-5 ساخته می‌شود که یک پنجره‌ی P^*L تصویر را از چپ به راست و از بالا به پایین اسکن می‌کند. همانطوری که پنجره‌ی نمونه برداری از چپ به راست روی یک خط حرکت می‌کند، هر مشاهده تعداد Q ستون همپوش با مشاهده‌ی قبل از خود دارد. زمانی که پنجره به پایان خط کنونی رسید، پنجره‌ی نمونه برداری به شروع خط باز می‌گردد و آن قدر به سمت پایین شیفت می‌کند تا به میزان M خط با پنجره‌ی بالایی خط قبل هم پوشانی

داشته باشد. هر مشاهده o_i شامل سطوح شدت پیکسل های نمونه برداری شده بوسیله ای پنجره است که به صورت بردار ستونی مرتب شده است.



شکل 5-3. تکنیک نمونه برداری برای یک ergodic HMM

شکل 6-3 داده های استفاده شده برای تمرین یک مدل ergodic و میانگین های توزیع های حالت های HMM را پس از تمرین نشان می دهد. HMM در این مورد دارای 8 حالت است با این فرض که پیشانی ، چشم ها ، بینی ، دهان ، چانه ، گونه ها و دو ناحیه ای کناری چهره هر یک ، یک حالت را در بر گیرند. نمایش میانگین های توزیع های حالت ها ممکن است در بدست آوردن چگونگی بخش بندی تصویر و اینکه کدام ویژگی ها یادگیری شوند ، کمک کند. به هر حال تصویر سمت راست شکل 6-3 هیچ ویژگی قابل شناسایی را نشان نمی دهد. بخشی از علت آن ، استفاده کردن از مدل ergodic بدون هیچ قیدی روی داده هاست که در نتیجه ای آن از اطلاعات ساختاری هیچ استفاده ای نمی شود. در بخش بعد مدل بالا - پایین ارائه می شود و نشان داده می

شود که چگونه از اطلاعات ساختاری استفاده می کند و توزیع های حالت ، ویژگیهای قابل تشخیص توسط انسان را ارائه می دهنند.

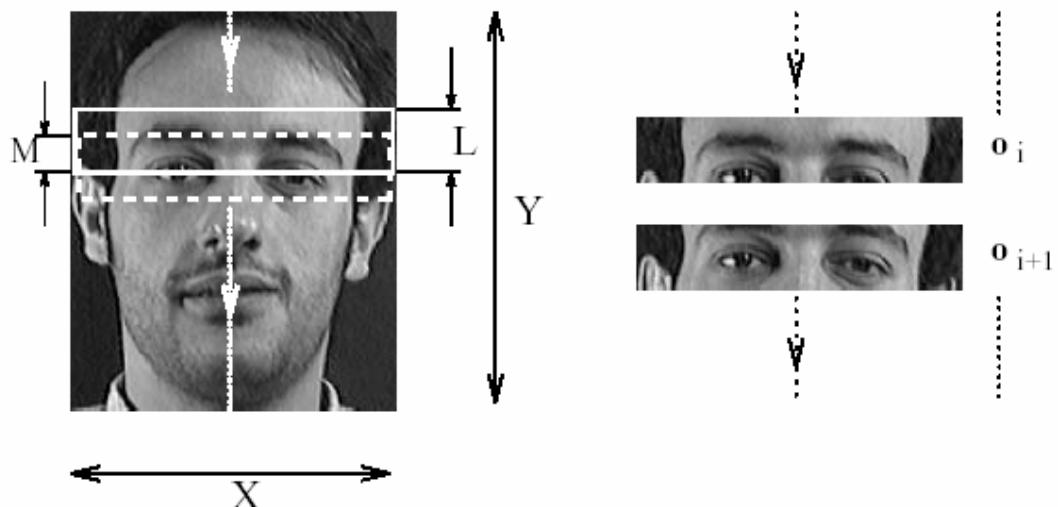


شکل 6-3 . داده تمرینی و میانگین های مدل برای ergodic HMM

HMM های بالا - پایین

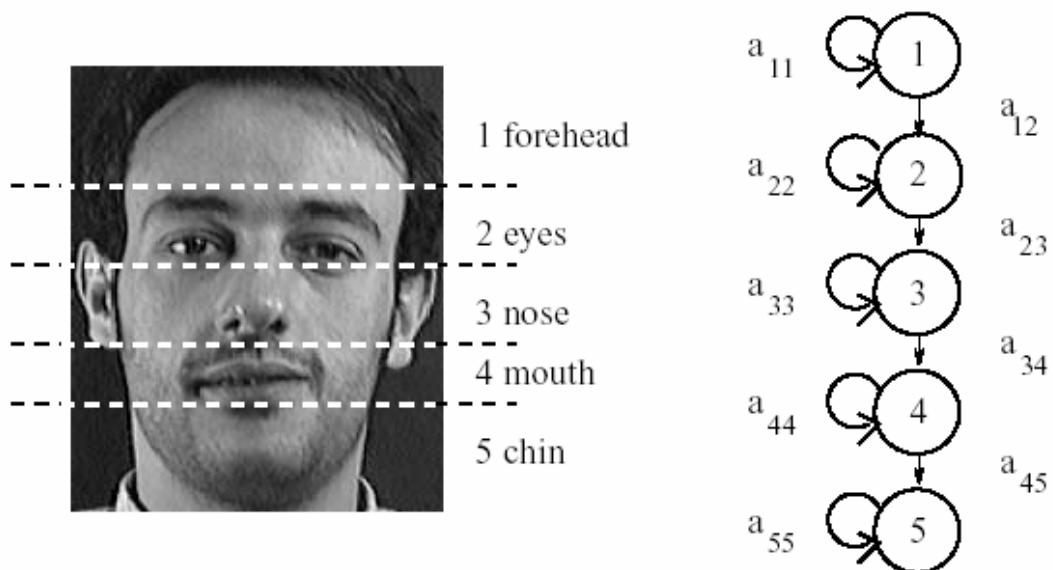
حالت های یک HMM در مدل ergodic به طور دلخواه می توانند ارتباط داده شوند که جزئیاتی از آن را در بخش قبل ملاحظه فرمودید. به هر حال ، برای کاربرد های تشخیص الگو ، بهتر است قید هایی بر روی انتقال حالت های مجاز گذاشته شود تا ویژگی های داده را reflect کند. به خصوص ، توبولوژی های HMM به نام چپ - راست¹ اغلب به کار می روند که این ویژگی را دارند که اندیس حالت بایستی به طور یکنواختی در زمان پردازش از طریق دنباله O مشاهده افزایش یابد. برای چهره ها ، ترتیب طبیعی ، پیمایش چهره از بالا به پایین است و بنابراین بالا - پایین ، یک طراحی طبیعی تر از چپ - راست است. شکل 7-3 ، یک تکنیک نمونه برداری با پیمایش پنجره ای چهره از بالا به پایین را نشان می دهد. دنباله O مشاهده ای O از تصویر Y^*X^* با استفاده از پنجره ای نمونه برداری X^*L پیکسلی که همان طور که در شکل نشان داده شده است X^*M پیکسل همپوشانی دارد.

¹ Left - right



شکل 7-3. تکنیک نمونه برداری برای یک HMM بالا - پایین

در شکل 3-8 یک HMM بالا - پایین را می بینید که پنج حالت دارد.



شکل 3-8 HMM . بالا - پایین با پنج حالت

استخراج ویژگی و بخش بندی داده های تمرین در این مدل در شکل 9-3 قابل مشاهده است. به سمت راست شکل نگاه کنید و ویژگی ها به دست آمده را ملاحظه کنید. مشاهده می شود که هر بخش ، یک ویژگی (بینی ، چشم ، پیشانی ، ...) را ارائه می دهد. مثلا در بخش سوم ، ویژگی چشم ها قابل تشخیص توسط انسان است.



شکل 9-3. داده های تمرینی بخش بندی شده و میانگین های حالت برای HMM بالا - پایین

قسمت سوم :

روش های دسته بندی

در تشخیص چهره

فصل چهارم :
تشخیص چهره با استفاده از
ماشین های بردار پشتیبان^۱

^۱ Support vector machines

ماشین های بردار پشتیبان ، یک الگوریتم تمرین برای یادگیری دسته بندی و قواعد رگرسیون از داده هاست. ماشین های بردار پشتیبان اولین بار در دهه ۱۹۶۰ توسط Vapnik برای دسته بندی پیشنهاد شد. آن ها از تئوری یادگیری آماری سرچشمه می گیرند و بر روی مینیمم سازی ریسک ساختاری شکل گرفته اند.

در این تئوری ، برای یک مسئله ای دسته بندی دو کلاسی ، هدف جداسازی دو کلاس بوسیله ای یک تابع است که از نمونه های موجود بدست می آید. به مثال شکل 4-1-a توجه کنید که در آن چندین دسته بند خطی ممکن ، وجود دارد که بتواند داده ها را جدا کند. اما فقط یکی از آن هاست (که در شکل 4-1-b نشان داده شده) که حاشیه (فاصله ای بین صفحه و نزدیکترین نقطه ای داده ای هر کلاس به صفحه) را افزایش می دهد. این دسته بند خطی ، صفحه ای جداکننده ای بهینه^۱ نامیده می شود. به مسئله ای جداسازی مجموعه ای بردارهای تمرین متعلق به دو کلاس مجزای

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l), \text{ where } \mathbf{x}_i \in R^n,$$

$$y_i \in \{-1, +1\}$$

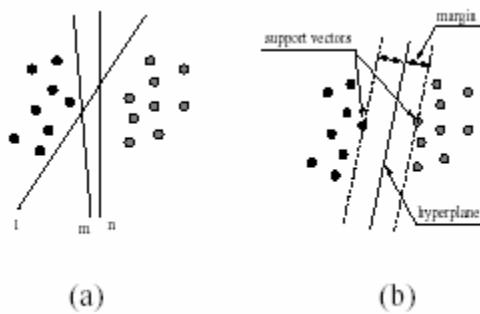
با صفحه ای

$$\mathbf{w}\mathbf{x} + b = 0.$$

^۱ Optimal separating hyperplane (OSN)

توجه کنید . به مجموعه \mathcal{S} بردارهایی مجزا شده ای بجهینه^۱ به وسیله ای صفحه گفته می شود ، اگر بدون خطأ جدا شوند و حاشیه ماکزیمم شود. یک صفحه ای کانونی یک قید برای پارامترهای w و b دارد که :

$$\min_{\mathbf{x}_i} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1.$$



شکل ۴-۱. دسته بندی دو کلاس به وسیله ای صفحه ها

l , m , n صفحه های دلخواه (a)

صفحه ای جداساز بجهینه با بیشترین حاشیه (b)

یک صفحه ای جداکننده در فرم کانونی بایستی قید زیر را برآورده کند :

$$y_i [(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1, i = 1, \dots, l \quad (4-1)$$

فاصله ای نقطه ای X از صفحه این گونه محاسبه می شود :

^۱ Optimally separated

$$d(\mathbf{w}, b; \mathbf{x}) = \frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|}$$

(4-2)

طبق تعریف ، حاشیه $\frac{2}{\|\mathbf{w}\|}$ است. بنابراین صفحه ای که به صورت بهینه ، داده را جدا می کند ، صفحه ای است که مقدار زیر را تا حد امکان کاهش بدهد :

$$\Phi(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2$$

(4-3)

راه حل برای مسئله ای بهینه سازی (معادله ای 4-3) تحت قیود (معادله ای 4-1) به وسیله ای نقطه ای saddle لگرانژ داده می شود

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i \{y_i [(\mathbf{w} \cdot \mathbf{x}_i) + b] - 1\}$$

(4-4)

که در آن α_i ضرایب لگرانژ هستند. حاصل مجبور است با توجه به \mathbf{b} ، \mathbf{w} مینیمم شود و با توجه به $\alpha_i \geq 0$ ماکزیمم شود. این دو قسمتی بودن موجب می شود تا مسئله ای بهینه سازی (معادله ای 4-4) به مسئله ای دو قسمتی آن تبدیل شود که حل آن آسان تر است. مسئله دو قسمتی به صورت (معادله ای 4-5) ارائه می شود.

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left\{ \min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha) \right\}$$

(4-5)

راه حل برای مسئله‌ی دو قسمتی این گونه است :

$$\bar{\alpha} = \arg \min_{\alpha} \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

(4-6)

با قیود

$$\alpha_i \geq 0, \quad i = 1, \dots, l$$

$$\sum_{i=1}^l \alpha_i y_i = 0$$

به ترتیب از بالا به پایین (4-7) و (4-8)

حل کردن معادله‌ی 6-4 با قیود (4-7 و 4-8) ، ضرایب لاگرانژ را حاصل می‌شود و OSH این گونه ارائه می‌شود

$$\bar{\mathbf{w}} = \sum_{i=1}^l \bar{\alpha}_i y_i \mathbf{x}_i$$

$$\bar{b} = -\frac{1}{2} \bar{\mathbf{w}} \cdot [\mathbf{x}_r + \mathbf{x}_s]$$

که \mathbf{x}_r ، \mathbf{x}_s بردارهای پشتیبانی هستند که قیود زیر را رعایت می‌کنند

$$\bar{\alpha}_r, \bar{\alpha}_s > 0, \quad y_r = 1, \quad y_s = -1$$

برای نقطه‌ی داده‌ای جدید X دسته‌بندی این گونه است :

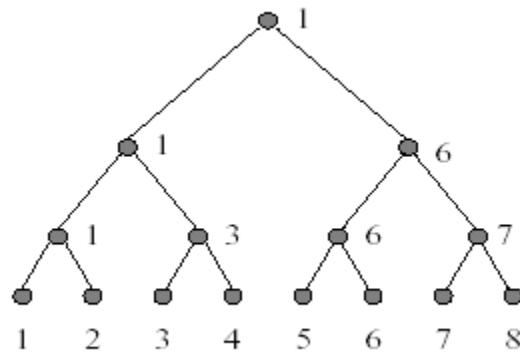
$$f(\mathbf{x}) = sign (\bar{\mathbf{w}} \cdot \mathbf{x} + \bar{b})$$

که البته این دسته‌بندی خطی بود.

تشخیص چند کلاسی

در بخش قبلی پایه‌های تئوری ماشین‌های بردار پشتیبان برای دسته‌بندی دو کلاس تشریح شد. یک تشخیص الگوی چند کلاسی می‌تواند به وسیله‌ی ترکیب ماشین‌های بردار پشتیبان دو کلاسی حاصل شود. به طور معمول دو دید برای این هدف وجود دارد. یکی از آنها استراتژی «یک در مقابل همه» برای دسته‌بندی بین هر کلاس و کلاس‌های باقیمانده است. دیگری استراتژی «یک در مقابل یک» برای دسته‌بندی هر جفت است. در شرایطی که دسته‌بندی اول به دسته‌بندی مبهم منجر می‌شود. دومی را برای سیستم تشخیص چهره وفق می‌دهیم. پیشنهاد می‌کنیم که یک درخت باینری پایین به بالا برای دسته‌بندی ساخته شود. فرض کنید هشت کلاس در مجموعه‌ی داده وجود دارد. درخت تصمیم در شکل ۴-۲ نشان داده شده است، جائیکه اعداد ۱ تا ۸ نشانگر هشت کلاس‌ها هستند. توجه کنید که در استفاده از این اعداد هیچ ترتیبی بکار نرفته است. برای یک چهره‌ی مورد تست، مقایسه‌ای بین هر جفت کلاس صورت می‌گیرد و یک کلاس به عنوان برنده‌ی مقایسه‌ی دو کلاس انتخاب می‌شود. کلاس‌های انتخاب شده از پایین

ترین سطح به یک سطح بالاتر می آیند و در آنجا دوباره مورد مقایسه واقع می شوند تا سرانجام یک کلاس در بالاترین نقطه‌ی درخت ظاهر شود.



شکل 2-4. ساختار درخت دودویی برای 8 کلاس تشخیص چهره. برای چهره‌ی تست ورودی ، آن چهره با هر دو جفت مقایسه می شود و برنده‌ها در سطحی بالاتر به همین صورت ، تست خواهند شد تا به نوک درخت برسیم که یک کلاس باقی بماند.

فصل پنجم :

تشخیص چهره با استفاده از

شبکه های عصبی

جستجو برای هوش مصنوعی (ماشین) یا سیستم هایی با رفتار انسانی چالشی فرا روی محققان شاخه های مختلف بوده است . بحث هایی بر روی تعریف یک سیستم هوشمند وجود دارد اما تنها سیستمی که به عنوان هوشمند مورد توجه قرار می گیرد ، مغز انسان است و این جاست که یک ایده‌ی مهندسی معکوس مطرح می شود . چرا باستی چیز هایی را که هم اکنون وجود دارند ، اختراع کنیم . در میانه های قرن بیستم ، ایده‌ای برای شبیه سازی مغز انسان از طریق طراحی پایین به بالای سیستم با شروع از نرون^۱ های به هم متصل و رسیدن به شبکه هایی که اعمال اصلی سیستم عصبی انسان را شبیه سازی می کنند ، مطرح شد .

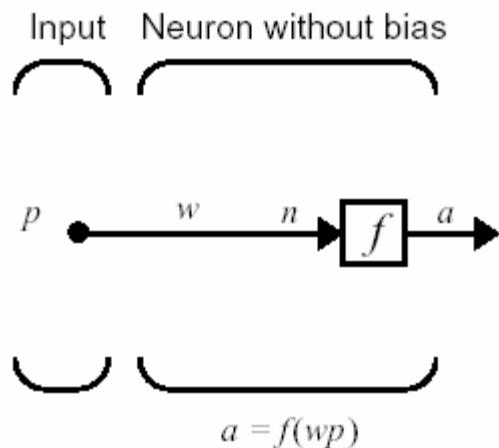
با گذر زمان ، شبکه های عصبی^۲ خودشان را به عنوان مدل محاسباتی مورد استفاده مطرح کردند . شبکه های عصبی معمولاً در انجام وظایف کامپیوتر های عام (عملیات با اعداد ، اعمال منطقی و ...) نادقيق هستند ، اما در زمینه هایی که کامپیوتر ها در آنجا به مشکل بر می خورند ، کارآیی بیشتری دارند مانند زمینه هایی که الگوریتم مشخصی وجود ندارد مانند پیش بینی ، تشخیص چهره و

مدل نرون

یک نرون با ورودی اسکالر و بدون بایاس در شکل 1-5 نشان داده شده است .

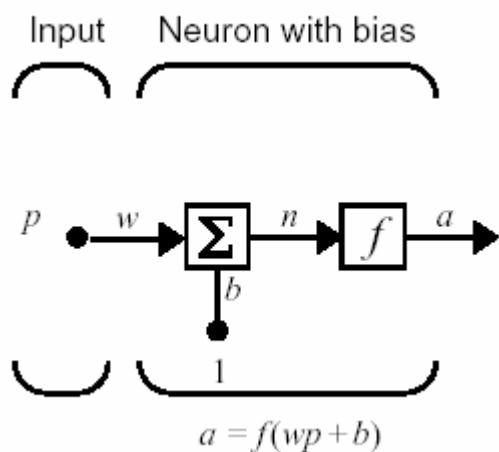
¹ neuron

² neural network



شکل 1-5. یک نرون با ورودی اسکالر و بدون بایاس

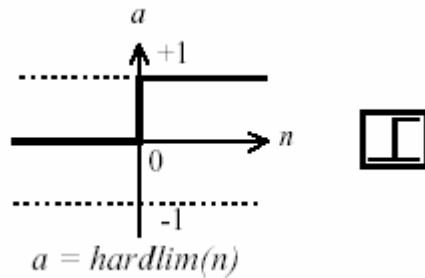
ورودی اسکالر p از طریق ارتباطی که مقدار آن را در وزن اسکالر w ضرب می کند به صورت حاصلضرب wp که اسکالر است ، منتقل می شود. در این مرحله ، ورودی وزن داده شده بخه عنوان آرگومان به تابع f منتقل می شود و این تابع خروجی a را تولید می کند. نرون شکل 2-5 یک مقدار بایاس b دارد. این مقدار به حاصلضرب wp اضافه شده است و تابع f را به سمت چپ محور X ها شیفت می دهد.



شکل 2-5. یک نرون با ورودی اسکالر و بایاس

توابع انتقال^۱

سه تابع انتقال که معمولاً مورد استفاده قرار می‌گیرد در شکل ۵-۳، ۵-۴ و ۵-۵ نشان داده شده است.

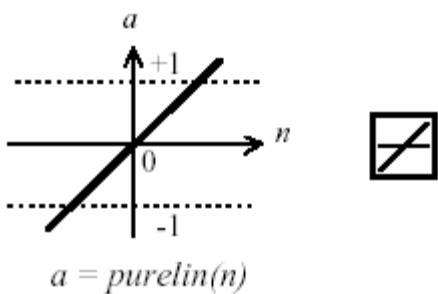


شکل ۵-۳. تابع انتقال hard-limit

تابع انتقال hard-limit، خروجی نرون را به مقادیر صفر و یک محدود می‌کند. اگر آرگومان ورودی تابع کمتر صفر بود، خروجی تابع مقدار صفر می‌باشد و آرگومان ورودی بزرگتر و یا مساوی صفر باشد، خروجی تابع یک می‌باشد. نرون‌هایی با این تابع انتقال در دسته بندی مورد استفاده قرار می‌گیرند.

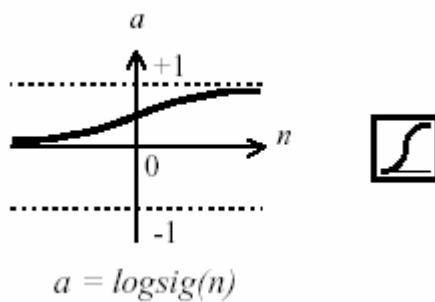
تابع انتقال خطی در شکل ۵-۴ نشان داده شده است. نرون‌هایی با این تابع انتقال در تقریب زننده‌های خطی استفاده می‌شوند.

^۱ Transfer function



شکل 4-5. تابع انتقال خطی

تابع سیگموئید در شکل 5-5 نشان داده شده است. مقدار خروجی این تابع در بازه $[0, 1]$ بین مقادیر صفر و یک محصور است. این تابع انتقال معمولاً در شبکه های انتشار عقب^۱ مورد استفاده قرار می گیرد.



شکل 5-5. تابع انتقال log-sigmoid

نرونی با ورودی برداری

یک نرون با بردار ورودی R عنصری در شکل 6-5 نشان داده شده است. بردار ورودی p p_1, p_2, \dots, p_R در بردار وزن W

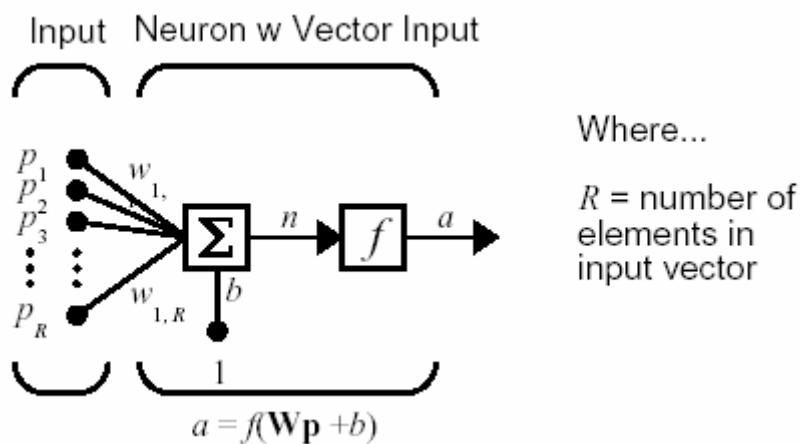
^۱ backpropagation

$$w_{1,1}, w_{1,2}, \dots w_{1,R}$$

به روش ضرب اسکالر ، ضرب می شود.

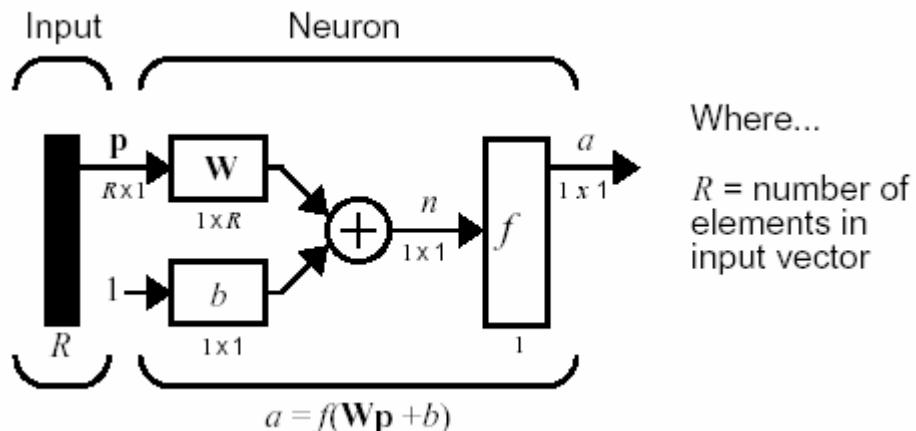
$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b$$

نتیجه‌ی این عملیات ضرب که یک مقدار عددی است ابتدا با مقدار بایاس جمع می شود. سپس این حاصل جمع به عنوان آرگومان ورودی به تابع f انتقال می یابد.



شکل 5-6 . یک نرون با بردار ورودی R عنصری

در شکل 7-5 ، نرون شکل 5-6 را با نشانه‌های سطح بالاتر می بینید.



Where...

R = number of elements in input vector

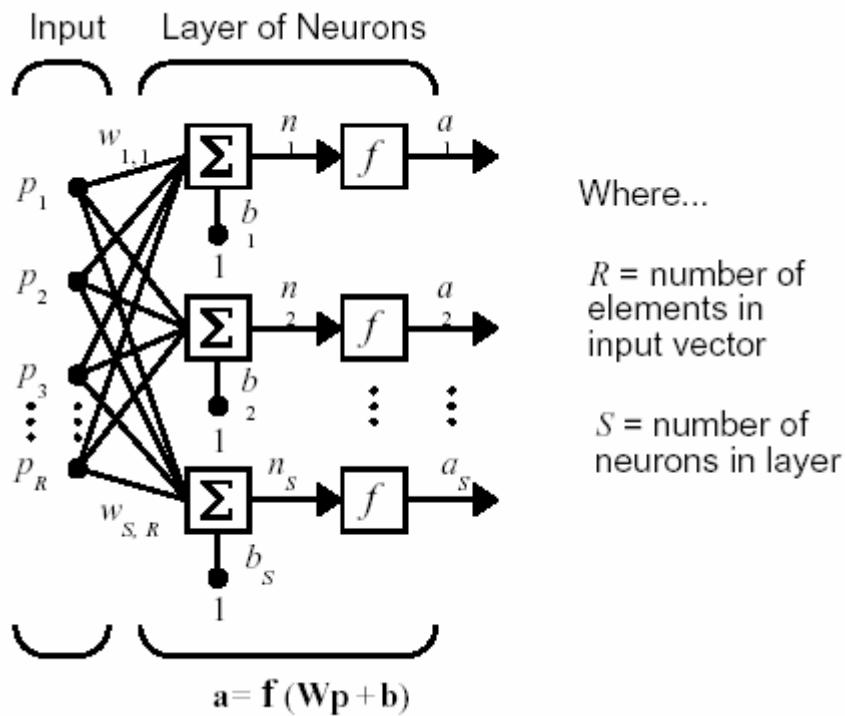
شکل 7-5 . یک نرون با بردار ورودی R عنصری (نشانه گذاری سطح بالاتر)

معماری های شبکه

دو یا چند نرون می توانند در یک لایه ترکیب شوند ، یک شبکه می تواند از یک یا چندین لایه داشته باشد. ابتدا یک لایه از نرون ها را بررسی می کنیم.

یک لایه از نرون ها

در شکل 8-5 ، یک شبکه ی یک لایه ای با R عنصر ورودی و S نرون را مشاهده می کنید.



Where...

R = number of elements in input vector

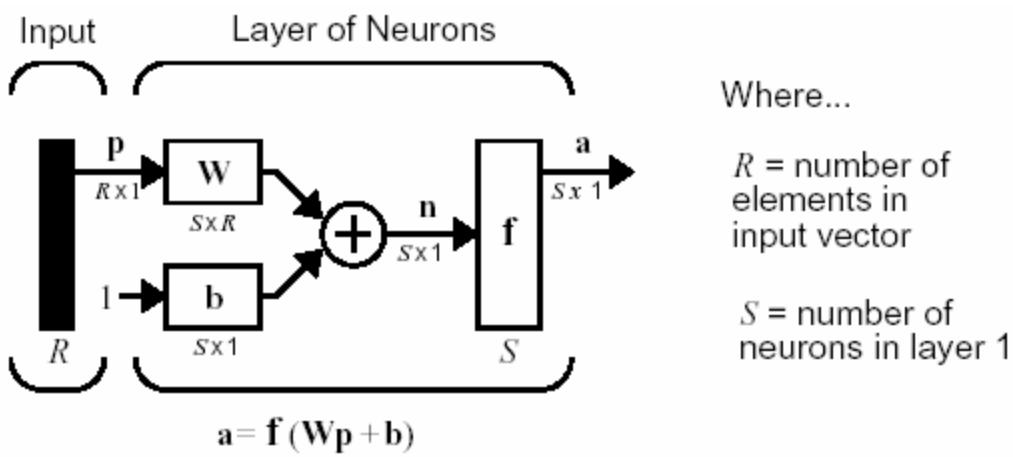
S = number of neurons in layer

شکل 8-5 . یک شبکه‌ی یک لایه‌ای با R عنصر ورودی و S نرون

در این شبکه، هر عنصر بردار ورودی p به هر نرون از طریق ماتریس وزن W مرتبط می‌شود. نرون i ام جمع کننده‌ای دارد که ورودی‌های وزن داده شده را با مقدار بایاس برای تولید مقدار خروجی $n(i)$ ، جمع می‌کند. $n(i)$ های بدست آمده از جمع کننده‌ها به عنوان ورودی به توابع انتقال فرستاده می‌شوند و در نهایت مقادیر a_i به عنوان خروجی لایه محسوب می‌شوند و بردار a را می‌سازند. الزامی وجود ندارد که تعداد ورودی‌ها به یک لایه با تعداد نرون‌های آن لایه برابر باشد. وزن‌های لایه در قالب ماتریس W قابل بیان است.

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

توجه کنید که اندیس سطر عناصر ماتریس \mathbf{W} به نرون مقصد وزن اشاره دارد و اندیس ستون عناصر به منبعی که ورودی وزن است، اشاره دارد. بنابراین اندیس های $W_{1,2}$ بر آن اشاره دارد که شدت سیگنال از عنصر ورودی دوم به اولین نرون، $W_{1,2}$ است. شبکه‌ی تک لایه‌ی مورد بحث با نشانه گذاری کوتاه‌تری در شکل 9-5 نشان داده شده است.

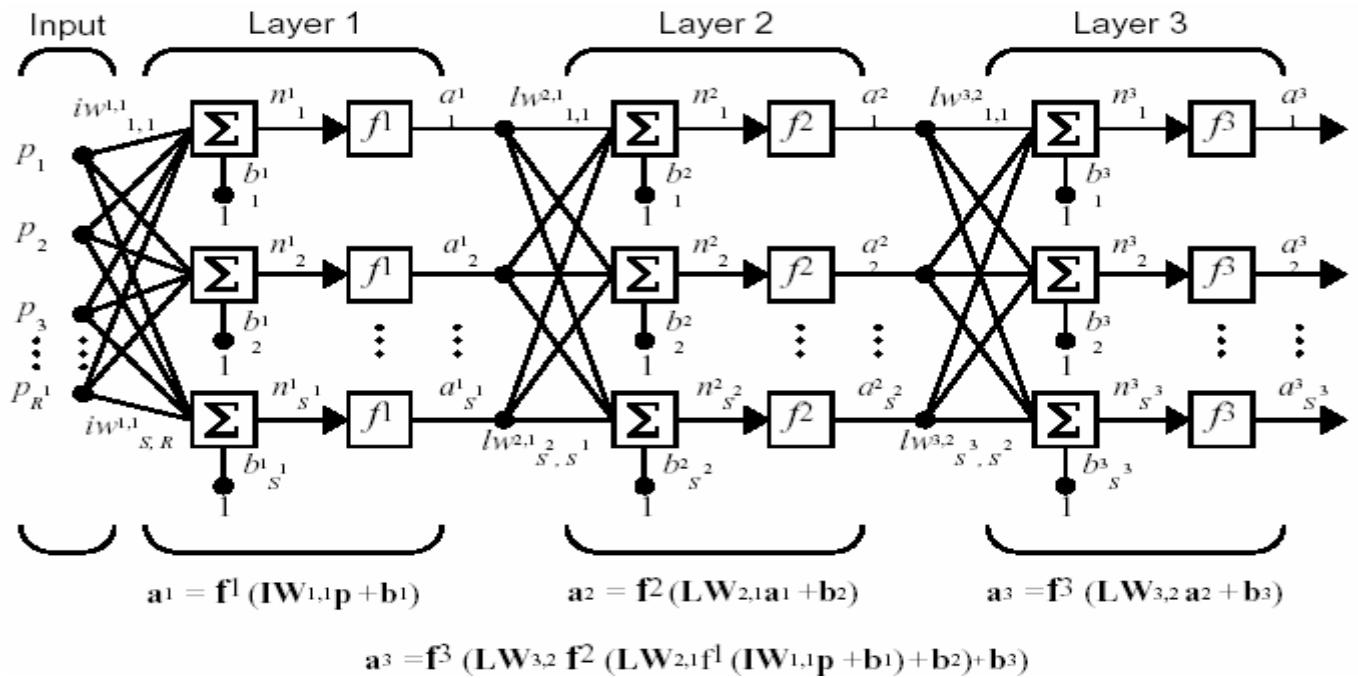


شکل 9-5. یک شبکه‌ی یک لایه‌ای با R عنصر ورودی و S نرون (با نشانه گذاری سطح بالاتر)

چندین لایه از نرون‌ها

یک شبکه می‌تواند چندین لایه داشته باشد. هر لایه یک ماتریس وزن \mathbf{W} ، یک بردار بایاس \mathbf{b} و یک بردار خروجی \mathbf{a} دارد. برای تشخیص بین ماتریس‌های وزن، بردار‌های خروجی و ...، برای هر یک از لایه‌ها، شماره‌ی لایه‌ی مربوطه را به متغیر

های مورد نظر اضافه می کنیم. می توان استفاده از این نشانه‌ی لایه را در شبکه‌ی سه لایه‌ای که در شکل ۱۰-۵ نشان داده شده است، ملاحظه نمود.



شکل ۱۰-۵. شبکه‌ی سه لایه‌ای

یادگیری شبکه‌های عصبی

شبکه‌های عصبی قادر به یادگیری از نمونه‌های ارائه شده است. در شبکه‌های عصبی، فرآیند یادگیری، تنظیم وزن‌های ارتباطات بین نرون‌هاست. از نقطه نظری دیگر، فرآیند یادگیری ممکن است به عنوان نوعی جستجو برای پیدا کردن مینیمم کلی تابع خطای شبکه تفسیر شود.

دو رویکرد برای این مسئله وجود دارد. اولی یادگیری نظارت یافته (یادگیری با آموزگار است.) در مرحله‌ی یادگیری ، برای هر نمونه‌ی یادگیری هر دوی ورودی و خروجی مورد نظر آن ورودی را داریم. بنابراین ، تعدادی جفت‌های تمرینی داریم (بردار‌های ورودی و خروجی ممکن است به ترتیب به عنوان « سوال » و « پاسخ صحیح » دیده شوند.) . این جفت‌ها به طور تکراری از شبکه گذرانده می‌شوند و در هر تکرار الگوریتم‌های یادگیری به کار برده می‌شوند و وزن‌ها تنظیم می‌شوند تا شبکه بتواند به ورودی‌ها ، پاسخ صحیح دهد (مثال نوعی این کلاس یادگیری ، الگوریتم « انتشار به عقب خطأ » است.) .

دومین رویکرد یادگیری بدون نظارت است. این فرم یادگیری تنها بردار ورودی را نیاز دارد و معمولاً نوعی از دسته بندی را بر طبق فاصله‌ی (مثل فاصله‌ی همینگ) یک بردار ورودی با بردارهای ورودی دیگر انجام می‌دهد. به هر حال نتایج این نوع یادگیری باقیستی آنالیز شود چرا که دسته بندی ممکن است نادرست باشد.

اکنون برای نمونه ، تعریفی رسمی از الگوریتم انتشار به عقب (برای یک شبکه‌ی سه لایه) ارائه می‌شود.

1. بردار ورودی به شبکه ارائه می‌شود.

2. feedforward انجام می‌شود با این هدف که هر نرون ، خروجی خود را با دنبال

$$o_i = \frac{1}{1 + e^{\left(\sum_{i=1}^n x_i w_i + b \right)}}$$

کردن بر روی نرون‌های لایه‌ی قبلی محاسبه کند.

3. خطای لایه خروجی برای هر نرون با استفاده از خروجی مطلوب (y_i) محاسبه می شود :

$$err_j^o = o_j(1 - o_j)(y_j - o_j)$$

4. خطای لایه میانی انتشار عقب می شود

$$err_i^h = h_i(1 - h_i) \sum_{j=1}^n err_j^o w_{ij}^o$$

5. با داشتن مقادیر محاسبه شده err_i^h و err_j^o وزن ها از لایه میانی به لایه خروجی و از وزوادی به لایه میانی به ترتیب طبق فرمول های

$$w_{ij}^o(t+1) = w_{ij}^o(t) + \alpha err_j^o h_i$$

و

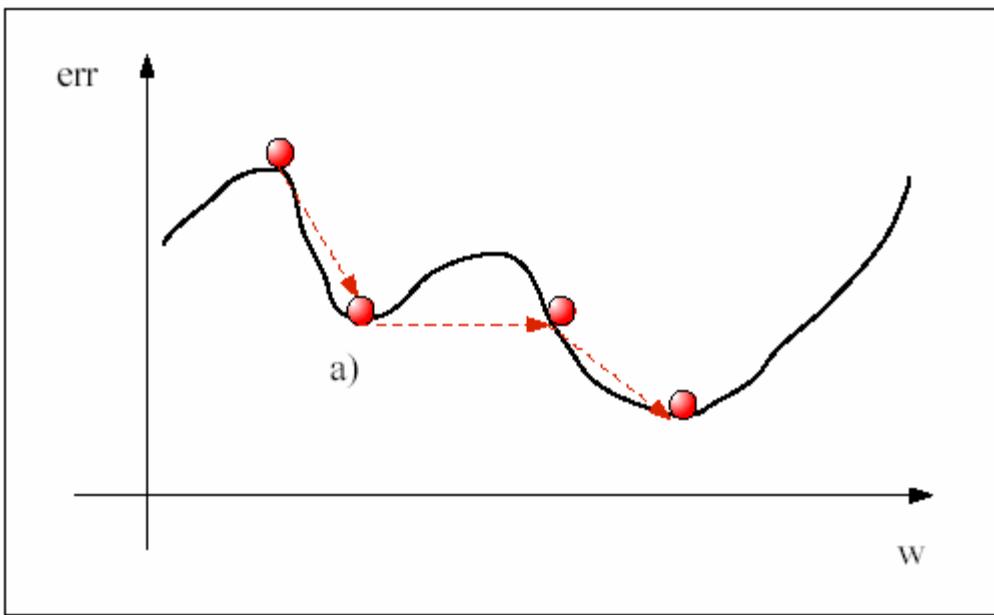
$$w_{ij}^h(t+1) = w_{ij}^h(t) + \alpha err_i^h x_i$$

می توانند تنظیم شوند که α ضریب یادگیری و x_i نرون i در لایه ورودی است .

6. تمامی مراحل قبل تا زمانی که خطای کلی شبکه

$$E = \frac{1}{2} \sum_{i=1}^n (y_i - o_i)^2$$

بر روی جفت های تمرین زیر سطح معینی نباشد ، ادامه می یابد.



شکل 11-5. مثالی از فرآیند انتشار عقب . خط چین ها جهت مشتق یک تابع خطرا در هر مرحله نشان می دهند.

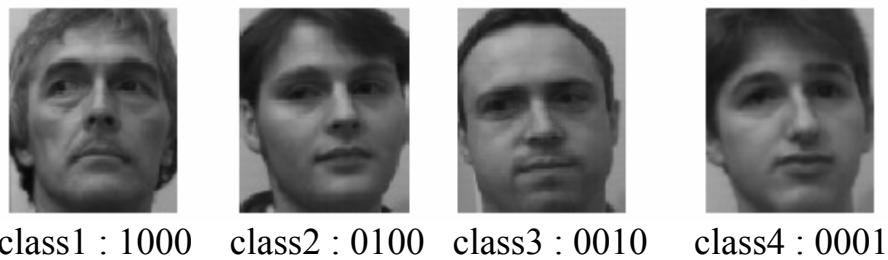
مسئله ای که در فرآیند یادگیری شبکه ممکن است رخ دهد قابلیت تعمیم شبکه است. انتشار به عقب مجموعه های تمرینی را به شبکه یاد می دهد اما در مواردی مجموعه های تمرین ، تمام محدوده ای مسئله را ارائه نمی دهد . در این وضعیت ما بخشی از مثال های مسئله را برای یادگیری به شبکه ارائه می کنیم و توقع داریم تا شبکه ای دانش را تعمیم دهد. این دلیلی است که مدل شبکه های عصبی موفق شمرده می شوند (به دلیل توانایی در تعمیم) . به عبارت دیگر ، در زمان ساخت شبکه ای برای تشخیص نوری اعداد دست نویس ، نیازی به فراهم کردن همه ای شکل های ممکن اعداد نیست و ممکن است مجموعه های تمرینی متشكل از تعدادی از نمونه های متدال تر مهیا کنیم و فرض می شود که شبکه یک « دید کلی » از هر عدد را یاد می گیرد به نسبت آن که هر نمونه را در حافظه ای خود نگه دارد. بنابراین ممکن است شبکه قادر به ارائه ی پاسخی صحیح به الگوهایی باشد که تا کنون مشاهده نکرده است.

دسته بندی با استفاده از شبکه های عصبی

یکی از کاربرد های شبکه های عصبی دسته بندی است . همانطورکه در فصل های قبل نیز گفته شده ، در فرآیند دسته بندی ، کلاسی که یک الگوی ورودی عضو آن است مشخص می گردد. در اینجا به نحوه‌ی سامان دهی شبکه‌ی عصبی و جفت‌های تمرین برای دسته بندی الگو ها می پردازیم.

اگر جفت‌های تمرین دسته بندی را داشته باشیم ، این بدان معناست که به عنوان ورودی ، الگو را داریم و به عنوان خروجی ، کلاسی که عضو آن است. ورودی را می توانیم به صورت یک بردار به شبکه ارائه کرد که در این صورت تعداد ورودی های شبکه برابر طول بردار خواهد بود. نکته‌ای که می ماند آنست که خروجی‌ها را چگونه طراحی کنیم تا دسته بندی را برآورده سازد.

برای رسیدن به این هدف ، فرمتی برای بردار خروجی معین می کنیم. این فرمت می تواند اینگونه باشد که به ازای هر کلاس ، یک عنصر در بردار خروجی داریم. اگر در پاسخ به یک ورودی ، مقدار یکی از این عناصر ، یک شود و عناصر دیگر مقدار صفر داشته باشند ، الگوی ارائه شده مربوط به کلاسی است که مقدار عنصر آن کلاس در بردار خروجی یک شده است . در شکل 12-5 این کد گذاری را برای تصاویر چهره‌ی ورودی به شبکه ملاحظه می کنید.



شکل 12-5 . کد گذاری خروجی برای کلاس های چهره

در این مورد ، چهار کلاس برای شناسایی چهار فرد داریم که هر کلاس به یک فرد منتسب است. شاید سر راست ترین گزینه برای تولید بردار از تصویر چهره‌ی دو بعدی به جهت ارائه به شبکه ، پیمایش سطر به سطر یا ستون به ستون تصویر باشد و از کنار هم قرار دادن سطر‌ها یا ستون‌ها ، یک بردار به دست آید. در تشخیص چهره ، این بردار ورودی که به شبکه عصبی ارائه می‌شود ، همان بردار ویژگی است. ویژگی‌هایی که در مراحل قبل فرآیند تشخیص چهره از تصویر استخراج شده‌اند و برای دسته‌بندی به شبکه‌ی عصبی ارائه می‌شوند . شبکه‌ی عصبی همانطور که قبلاً بحث شد ، فرآیند یادگیری را پشت سر می‌گذراند تا برای دسته‌بندی چهره‌ها آماده شود.

منابع

- [1] Goldstein, A. J., Harmon, L. D., and Lesk, A. B., Identification of human faces", Proc. IEEE 59, pp. 748-760, (1971).
- [2] Haig, N. K., "How faces differ - a new comparative technique", Perception 14, pp. 601-615, (1985).
- [3] Rhodes, G., "Looking at faces: First-order and second order features as determinants of facial appearance", Perception 17, pp. 43-63, (1988).
- [4] Kirby, M., and Sirovich, L., "Application of the Karhunen-Loeve procedure for the characterization of human faces", IEEE PAMI, Vol. 12, pp. 103-108, (1990).
- [5] Sirovich, L., and Kirby, M., "Low-dimensional procedure for the characterization of human faces", J. Opt. Soc. Am. A, 4, 3, pp. 519-524, (1987).
- [6] Turk, M., and Pentland, A., "Eigenfaces for recognition", Journal of Cognitive Neuroscience, Vol. 3, pp. 71-86, (1991).
- [7] Ilker Atalay , " Face recognition using eigenfaces ", M. Sc. THESIS , (January, 1996).
- [8] F. Samaria, “*Face Recognition Using Hidden Markov Models*” , PhD thesis, University of Cambridge, 1994.
- [9] Guodong Guo, Stan Z. Li, and Kapluk Chan , “Face Recognition by Support Vector Machines ”
- [10] L.R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceeding of the IEEE , 77(2):257-286, 1989.
- [11] L.E. Baum. “An inequality and associated maximization technique in statistical estimation for probabilistic function of Markov processes”. Inequalities , III:1-8 , 1972.
- [12] Y. He and A. Kundu. 2-D shape classification using Hidden Markov Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(11):1172-1184, 1991.
- [13] A. Kundu , Y. He and P.Bahl. Recognition of handwritten word: first and second order Hidden Markov Model based approach. Pattern Recognition, 22(3):283-297, 1989.
- [14] E. Levin and R.Pieraccini. “Dynamic Planar wraping for optical character recognition” , Proceedings of the International Conference on Acoustics , Speech and Signal Processing. III:149-152, 1992.
- [15] O.E. Agazzi, S. Kuo, E. Levin, and R. Pieraccini. “ Connected and degraded text recognition using planar hidden markov models ” , Procedings of International Conference on Acoustics , Speech and Signal Processing, V:113-16 , 1993.
- [16] S. Kuo and O.E. Aggazi. “Machine vision for keyword spotting using pseudo 2D Hidden Markov Models” , Proceedings of the International Conference on Acoustics , Speech and Signal Processing, V:81-84, 1993.
- [17] Dominik Ramik , “Modular Neural Networks” , Thesis, UNIVERSITY OF OSTRAVA FACULTY OF SCIENCE

سایت ها

1. www.face-rec.org
2. www.Mathworks.com