

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 مارکف 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 :

تشخیص الگو و

تشخیص 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 ی 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 ی خاص و تمیز دادن آن از یک تعداد فراوان از مدل های چهره ی 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 کاملاً آشکار نیست. تشخیص چهره ی انسان برای بیش از بیست سال مورد مطالعه قرار گرفته است. توسعه ی یک مدل محاسباتی برای تشخیص چهره کاملاً 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 ی 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 شیء 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 ی اصلی تقسیم می شوند:

تشخیص آیم 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 آیت‌های انتزاعی^۱ از طرف دیگر، یک بحث قدیمی یا یک راه حل برای مسئله، قابل تشخیص است. این فرآیند شامل تشخیص آیت‌های انتزاعی است و می‌توان آن را تشخیص انتزاعی خواند.

در واقع تشخیص انسان، یک پرسش تخمین زدن احتمال نسبی ای است که یک داده‌ی ورودی به یکی از مجموعه‌های جمعیت‌های آماری شناخته شده، که وابسته به تجربیات گذشته است، می‌تواند مرتبط باشد. بنابراین مسئله‌ی تشخیص الگو ممکن است به عنوان یک مسئله‌ی تشخیص اختلاف داده‌ی ورودی با جمعیت‌ها به وسیله‌ی جستجو‌ی ویژگی‌ها یا صفات ثابت در میان اعضای یک جمعیت مورد توجه قرار گیرد.

الگوها و کلاس‌های الگوها

تشخیص الگو می‌تواند به عنوان دسته‌بندی داده‌های ورودی در کلاس‌های شناخته شده به وسیله‌ی استخراج ویژگی‌های مهم یا صفات داده تعریف شود.

یک کلاس الگو، یک دسته‌ی متمایز شده به وسیله‌ی برخی صفات و ویژگی‌های مشترک است. ویژگی‌های یک کلاس الگو، صفات نوعی هستند که بین همه‌ی الگوهای متعلق به آن کلاس مشترک هستند. این ویژگی‌ها اغلب به عنوان ویژگی‌های اینترست^۲ شناخته می‌شوند. ویژگی‌هایی که تفاوت‌های بین کلاس‌های الگو را بیان می‌کنند اغلب به عنوان ویژگی‌های اینترست^۳ شناخته می‌شوند.

^۱ Conceptual items

^۲ intraset

^۳ interset

یک الگو ، توصیفی از یکی از اعضای دسته است که ارائه دهنده ی کلاس الگو می باشد. برای راحتی ، الگوها معمولا به وسیله ی یک بردار نمایش داده می شوند. مانند :

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

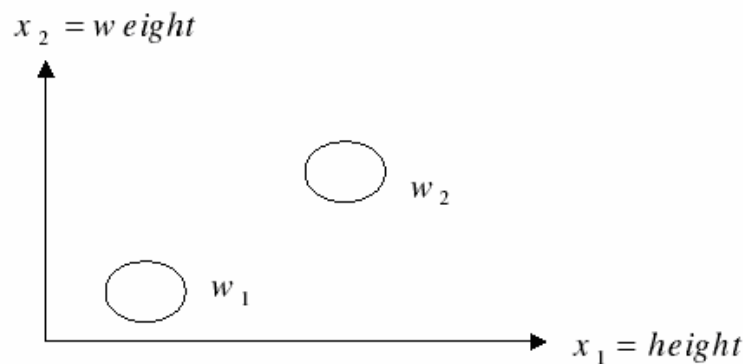
که در آن عنصر X_i یک ویژگی آن الگو را ارائه می کند. اغلب مفید است که یک بردار الگو به عنوان یک نقطه در فضای اقلیدسی n بعدی دیده شود.

مسائل اساسی در طراحی سیستم تشخیص الگو

به طور کلی طراحی یک سیستم تشخیص الگو چندین مسئله ی اصلی را در بر می گیرد :

- اول از همه ، ما بایستی در مورد نمایش داده های ورودی تصمیم بگیریم. چرا که بایستی اشیائی که قرار است مورد تشخیص قرار گیرند ، اندازه گیری شوند. این یک مسئله ی sensing است. هر کمیت اندازه گیری شده یک ویژگی الگو یا شیء را توصیف می کند و مجموعه ی این ویژگی ها در یک بردار قرار می گیرند. این بردار الگو که داده ی ورودی را توصیف می کند ، بایستی قالب

بندی شود. بردارهای الگو همه یا بخشی از اطلاعات اندازه گیری شده موجود را در مورد الگوها در بر می گیرد. مجموعه ی الگوهای متعلق به کلاسی یکسان به گروهی از نقاط در یک فضای اندازه گیری شده نگاشت می شوند. یک مثال ساده از این مورد در شکل 1 برای دو کلاس w_1 و w_2 نشان داده شده است.



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

- دومین مسئله در تشخیص الگو، استخراج ویژگیها یا صفات خاصی از داده ی ورودی دریافته شده و کاهش ابعاد بردارهای الگوست. این مورد اغلب به عنوان مسئله ی پیش پردازش و استخراج ویژگی معرفی می شود. عناصر ویژگیهای (اینتراست) برای همه ی کلاس های الگو مشترک هستند می توانند حذف شوند. اگر یک مجموعه ی کامل از ویژگیهای تشخیص برای هر کلاس از داده های 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.

- مسئله ی سوم در طراحی سیستم تشخیص الگو تعیین رویه های تصمیم بهینه است که در فرآیند شناسایی و دسته بندی مورد نیاز واقع می شود. پس از آنکه داده های مشاهده شده از الگوها جمع آوری شد و در فرم نقاط الگو یا بردارهای اندازه گیری در فضای الگو بیان شد ، ما ماشینی را می خواهیم 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 می تواند شامل M ناحیه باشد که هر کدام نقاط الگوی یک کلاس را در بر می گیرد. مسئله ی تشخیص حالا می تواند به عنوان تولید کننده ی محدوده های تصمیم که M کلاس را روی بردارهای اندازه گیری از همدیگر جدا می کند در نظر گرفته شود. به طور کلی ، این محدوده های تصمیم به وسیله ی توابع تصمیم تعیین می شوند.

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 ، حدس هایی مستدل از فرم های تابع تصمیم می توان داشت. در این مورد محدوده های 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 ماشین های تشخیص الگو با استفاده از یک رویه ی یادگیری یا تمرین دادن طراحی بهتری خواهند داشت. به صورت ابتدائی ، توابع تصمیم موقت فرض می شوند 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 بسپاریم که تمرین و یادگیری فقط در طول فاز طراحی سیستم تشخیص الگو انجام می شوند. هنگامی که نتایج قابل قبول با مجموعه ی الگوهای تمرینی به دست آمد ، سیستم برای وظیفه ی اجرایی واقعی خود بر روی نمونه های محیطی به کار گرفته می شود. کیفیت کارآئی تشخیص به طور گسترده ای به وسیله ی تشابه الگوهای تمرینی و داده های واقعی که سیستم در طول عملیات مواجه خواهد شد ، تعیین می شود.

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

در بسیاری موارد ، الگوهای نماینده از هر کلاس موجود هستند. در این وضعیت ها ، تکنیک های تشخیص الگوی نظارت شده ، کاربردی هستند. پایه های این رویکرد ، مجموعه ای از الگوهای تمرینی شناخته شده برای دسته بندی و پیاده سازی یک رویه ی 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 ی بدون نظارت کاربرد دارند. همانطور که در بالا بیان شد ، تشخیص الگوی نظارت یافته به وسیله ی این موضوع که دسته ی صحیح هر الگوی تمرینی مشخص است ، معرفی می شوند. در مورد بدون نظارت ، به هر حال ، با مسئله ی یادگیری در کلاس های الگوی ارائه شده

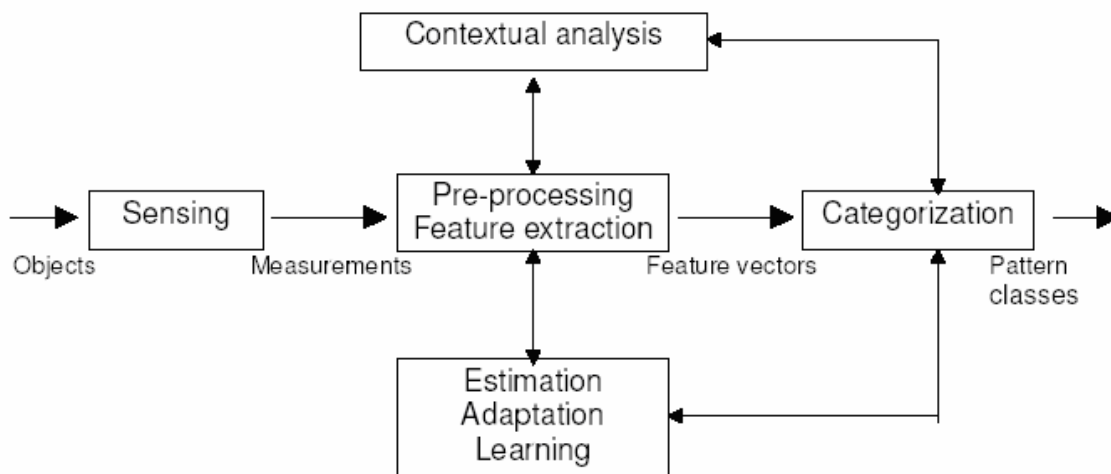
¹ Supervised

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 ی 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 بندی 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

large pattern deviations انعطاف پذیر باشد و نیز قابلیت خود تنظیمی را داشته باشد، ما با مسئله ی تطبیق دادن مواجه شده ایم.

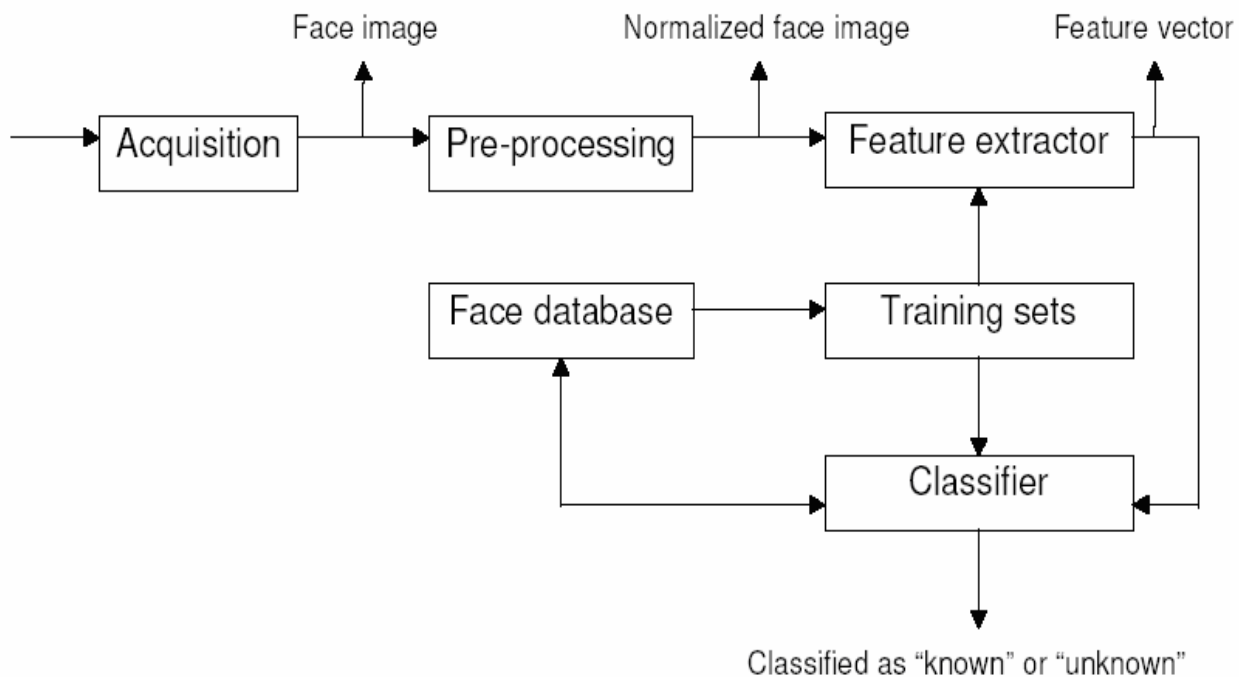
تشخیص چهره

تشخیص چهره یک عمل تشخیص الگو است که به طور خاص بر روی چهره ها انجام می شود. این عمل عبارتست از دسته بندی یک چهره به عنوان « شناخته شده » و « ناشناس » پس از مقایسه با چهره های ذخیره شده ی افراد شناخته شده.

مدل های محاسباتی تشخیص چهره بایستی پاسخگوی چندین مسئله ی دشوار باشند. این سختی از آنجا ناشی می شود که چهره ها بایستی به گونه ای ارائه شوند که اطلاعات موجود در چهره را برای تشخیص یک چهره ی خاص از دیگر چهره ها به بهترین نحو بکار برند. در این مورد ، چهره ها باعث یک مسئله ی دشوار می شوند چرا که همه ی چهره ها از آنجا که مجموعه ویژگی های یکسانی مانند چشمان ، بینی و دهان دارند ، مشابه با یکدیگر هستند.

کلیات یک سیستم تشخیص چهره عام

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



شکل 3-1 - یک سیستم تشخیص چهره ی عام

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

- **ماژول دریافت** . این نقطه ی ورود به فرایند تشخیص چهره است. این ماژولی است که تصویر چهره ی مورد نظر به سیستم ارائه می شود. به عبارت دیگر، در این ماژول ، از کاربر درخواست می شود تا یک تصویر چهره به سیستم تشخیص چهره ارائه کند. یک ماژول دریافت می تواند تصویر چهره را از چندین محیط متفاوت تحویل بگیرد : یک تصویر چهره ممکن است یک فایل تصویر بر روی یک دیسک مغناطیسی باشد ، ممکن است به وسیله ی 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 بینایی¹ ، تصاویر چهره نرمال سازی می شوند و اگر نیاز بود ، برای افزایش کارائی 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 ی تصویر.** این کار معمولاً انجام می شود تا اندازه ی تصویر گرفته شده به یک اندازه تصویر پیش فرض مانند 128×128 تغییر کند ، این اندازه ی تصویر همان اندازه ی 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 کارائی تشخیص چهره انجام می شود که گستره ی **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 ی بهتری بدهند. فیلترینگ بالا گذر بر روی جزئیاتی مانند لبه ها تاکید دارند که در نتیجه کارائی 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 ، مازول 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 مازول پیش - پردازش وجود چرخش یا انتقال را مشخص کند و اگر ممکن بود ، به نرمال سازی انتقال ها و چرخش ها در مکان سر بپردازد.

7. نرمال سازی نور پردازی. تصاویر چهره ی گرفته شده در شرایط نوری متفاوت می توانند کارائی 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.

- مازول 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 مازول استخراج ویژگی فرستاده می شود. به عبارت

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

• **ماژول دسته بندی.** در این مازول ، با کمک یک دسته بند الگو ، ویژگیهای استخراج شده از تصویر چهره با آنهایی که در کتابخانه ی چهره (یا پایگاه داده ی چهره) ذخیره شده اند مقایسه می شود. پس از انجام این مقایسه ، تصویر چهره به عنوان « شناخته شده » و « نا شناس » دسته بندی می شود.

• **مجموعه ی تمرین.** مجموعه های تمرینی در طول « فاز یادگیری » فرآیند تشخیص چهره استفاده می شوند. مازول های استخراج ویژگی و دسته بندی به منظور بدست آوردن کارائی 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. مازول دسته بندی به طور مستقیم از پایگاه داده ی چهره استفاده می کند.

ماژول های استخراج ویژگی و دسته بندی ، دو مازولی هستند که وجه تمایز بین بیشتر سیستم های تشخیص چهره ی متفاوت را باعث می شوند. تقریباً می توان گفت که تفاوت کارائی سیستم های تشخیص چهره ، در امر تشخیص چهره ها از کارائی این دو مازول آن ها متأثر است. در بخش های بعدی به چهار روش از روش های متفاوتی که سیستم های تشخیص چهره در این دو مازول 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 حداکثر 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 ی تصاویر چهره هاست سپس می توان با استفاده از این اطلاعات به کدگذاری و مقایسه ی تصاویر چهره ی افراد پرداخت.

به زبان ریاضی ، اجزای اصلی توزیع چهره ها ، یا بردار - ویژه های ماتریس کواریانس مجموعه ی 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.

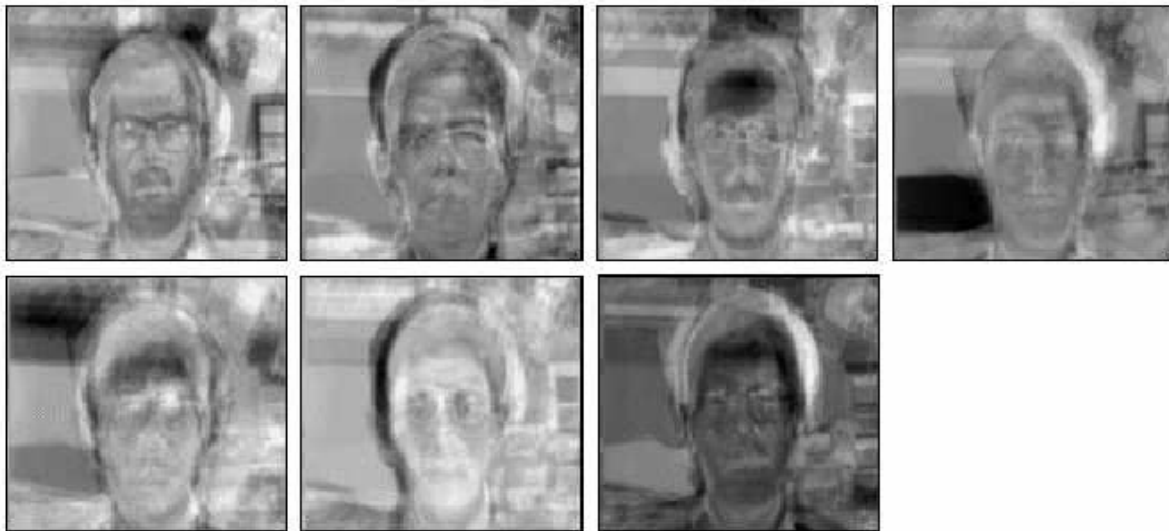
بردار ویژه ها می توانند به عنوان مجموعه ای از ویژگیها فرض شوند که باهم اختلاف بین تصاویر چهره ها را مشخص می کنند. در اینجا چون بردار ویژه ها در مورد تصاویر چهره هستند ، چهره - ویژه¹ نامیده می شوند.

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) مجموعه ی تمرین تصاویر چهره به صورت نمونه (b) تصویر چهره ی میانگین
مجموعه ی تمرین



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

هر چهره ای می تواند دقیقا به وسیله ی ترکیب خطی چهره - ویژه ها ارائه شود. همچنین هر چهره می تواند با استفاده از « بهترین » چهره - ویژه ها ، آنهایی که بیشترین مقادیر ویژه را دارند ، تقریب زده شود. M چهره - ویژه ی بهترین ، یک زیر فضای M بعدی را به وجود می آورد که ما « فضای چهره » ی همه ی تصاویر ممکن می نامیم.

kirby و Sirovich [4,5] تکنیکی را برای ارائه ی تصاویر چهره ها به طور کارا با استفاده از آنالیز جزء اصلی توسعه دادند آن ها بهترین مختصات سیستم را برای فشرده سازی تصویر محاسبه کردند که هر محور مختصات واقعا یک تصویر است که آن را « تصویر - ویژه » نامیدند. آن ها بیان کردند ، حداقل در تئوری ، هر مجموعه ای از تصاویر به صورت تقریبی می تواند به وسیله ی ذخیره کردن یک مجموعه ی کوچک از وزن ها برای هر چهره و مجموعه ای کوچک از تصاویر استاندارد (تصویر - ویژه ها) بازسازی

شود. وزن های توصیف کننده ی هر چهره به وسیله ی تصویر کردن¹ 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 ی یک سیستم تشخیص چهره ی مبتنی بر رویکرد چهره - ویژه پیشنهاد شده را دنبال خواهیم کرد. آن ها بیان کردند که تعداد زیادی از تصاویر چهره به وسیله ی مجموع وزن داده شده ی مجموعه ی 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 ی تجربه و تشخیص چهره ها به وسیله ی مقایسه ی وزن های ویژگی های مورد نیاز برای بازسازی تقریبی آنها با وزنه های مرتبط با چهره های شناخته شده باشد. بنابراین ، هر چهره به وسیله ی مجموعه ی کوچکی از ویژگی ها یا وزن های تصویر - چهره ی مورد نیاز برای توصیف و بازسازی آن ها مشخص می شود.

کلیات سیستم تشخیص چهره

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

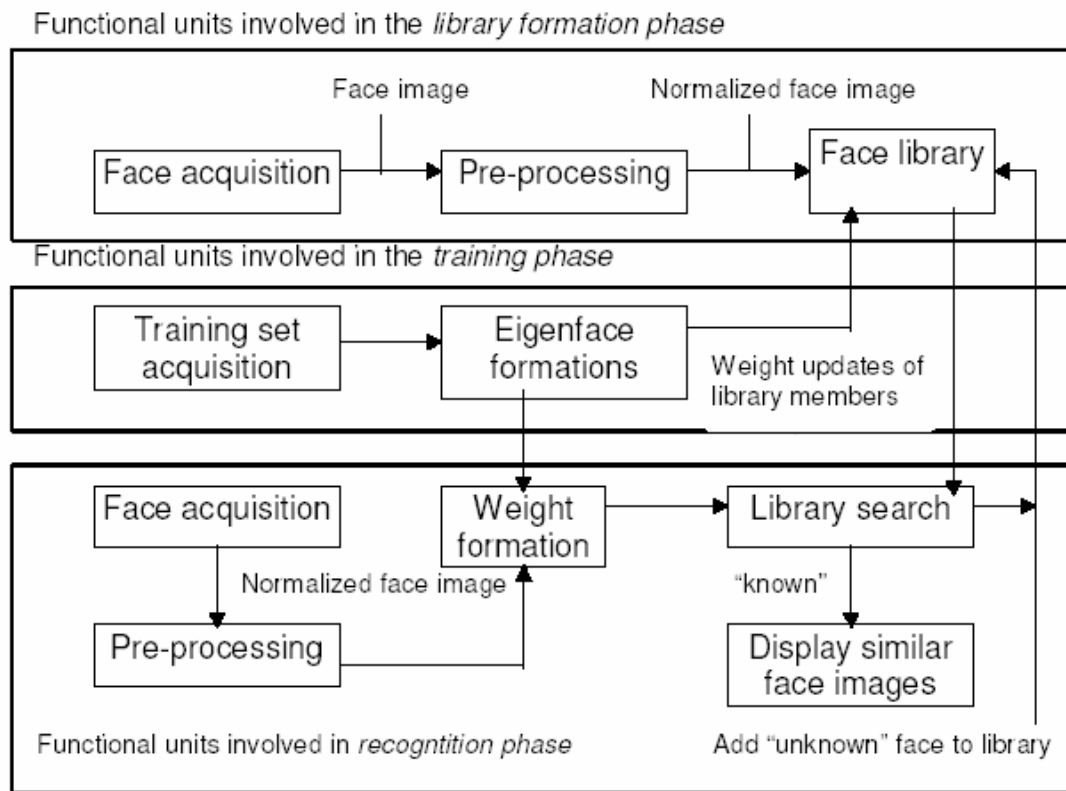
¹ Projecting

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

● **فاز تمرین :** پس از اضافه کردن تصاویر چهره ها به کتابخانه ی چهره ، سیستم برای انجام تمرین و تشکیل چهره - ویژه ها آماده است. تصاویر چهره ای که می خواهیم در تمرین حضور داشته باشند از کتابخانه ی چهره انتخاب می شوند. از آنجائی که عناصر کتابخانه ی چهره نرمال شده هستند ، هیچگونه عمل پیش - پردازشی در این مرحله صورت نمی پذیرد. پس از انتخاب مجموعه ی تمرین ، چهره - ویژه ها تشکیل و برای استفاده های بعد ، ذخیره می شوند. چهره - ویژه ها از روی مجموعه ی تمرین محاسبه می شوند و فقط M تصویر مربوط به بیشترین مقادیر ویژه نگهداری می شوند. این M تصویر چهره ، « فضای چهره » ی M بعدی را نشان می دهند. همان طور که چهره های جدید تجربه می شوند ، چهره ویژه ها بروز رسانی و یا دوباره محاسبه می شوند. اوزان مربوطه در فضای وزن M بعدی برای هر عضو کتابخانه محاسبه می شود (به وسیله ی تصویر کردن تصویر چهره ی آن بر روی « فضای چهره » ای که به وسیله ی چهره - ویژه ها 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 ی تمرین و به وجود آوردن بردارهای وزن عناصر تصویر چهره ، اکنون سیستم آماده ی فرآیند تشخیص چهره است. کاربر فرآیند تشخیص را با انتخاب یک تصویر آغاز می کند. بر اساس درخواست کاربر و اندازه ی تصویر دریافتی ، مراحل پیش - پردازش برای نرمال سازی این تصویر با مشخصات کتابخانه ی 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 ی 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 ی جدید به همراه بردار وزن آن می تواند به کتابخانه ی چهره اضافه شود تا بعدا مورد استفاده قرار گیرد (یادگیری برای تشخیص).



شکل 2.3 - دیاگرام بلوکی سیستم تشخیص چهره ی ارائه شده

محاسبه ی چهره - ویژه ها

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

تصاویر چهره ها، به خاطر شباهت در پیکربندی کلی، به طور تصادفی در این فضای بزرگ توزیع نمی شوند و بنابراین به وسیله ی یک زیر فضای به نسبت کم بعد قابل

توصیف هستند. ایده ی 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 می کنند که ما « فضای چهره » می نامیم. طول هر بردار N به توان 2 است ، که یک تصویر $N*N$ را توصیف می کند ، که یک ترکیب خطی از تصاویر چهره ی اصلی است. چون این بردارها ، بردارهای ویژه ی ماتریس کواریانس مربوط به تصاویر چهره ی اصلی هستند و چون آنها در ظاهر شبیه چهره هستند ، ما آنها را چهره - ویژه می نامیم.

تعاریف :

ماتریس $N*N$ با نام A ، بردار ویژه ی X و مقدار ویژه ی λ را دارد اگر

$$AX = \lambda X.$$

و

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

که اگر بسط داده شود یک چند جمله ای درجه ی N ام با متغیر λ است که ریشه های آن مقادیر ویژه هستند. این اثبات می کند که معمولا N مقدار ویژه وجود دارند. مقادیر ویژه ی مساوی از ریشه های مضاعف بر می آیند که « degenerate » نامیده می شود.

یک ماتریس متقارن نامیده می شود اگر مساوی با ترانهاده ی خود باشد

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

آن متعامد نامیده می شود اگر ترانهاده ی با معکوس آن برابر باشد

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

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

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

قضیه : مقادیر ویژه ی یک ماتریس متقارن حقیقی ، همگی حقیقی هستند. در مقابل ، مقادیر ویژه ی یک ماتریس نامتقارن حقیقی ممکن است مقادیر حقیقی را شامل شود اما همچنین ممکن است مقادیر مختلط را نیز در بر گیرد. مقادیر ویژه ی یک ماتریس نرمال با مقادیر ویژه ی « nondegenerate » کامل و متعامد هستند پس از ارائه ی توضیحاتی بر روی واژه هایی که قصد داریم در محاسبه ی چهره - ویژه ها از آن ها استفاده کنیم ، می توانیم به فرآیند پیدا کردن این چهره - ویژه ها بپردازیم.

فرض کنید ، مجموعه ی تصاویر چهره $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ باشند و میانگین مجموع به وسیله ی فرمول زیر تعریف می شود :

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

بردار اختلاف هر چهره از میانگین اینگونه محاسبه می شود

$$\Phi_l = \Gamma_l - \Psi$$

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

این مجموعه بردار های خیلی بزرگ موضوع کاری آنالیز جزء اصلی است که یک مجموعه از بردار های orthonormal U_n ، که بهترین توصیف از توزیع داده ها باشند، را جستجو می کند. بردار K ام ، U_k ، انتخاب می شود که

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

یک ماکزیمم است که

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

بردار های U_k و اسکالر های λ_k به ترتیب بردار های ویژه و مقادیر ویژه ی ماتریس کواریانس

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

که ماتریس $A = [\Phi_1 \ \Phi_2 \ \dots \ \Phi_M]$. ماتریس کواریانس C ، به هرحال ماتریس متقارن حقیقی $N^2 * N^2$ است و تعیین N^2 بردار ویژه و مقدار ویژه یک کار حجیم 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 ($M < N^2$) آنجا $M-1$ به نسبت N^2 بردار ویژه ی با معنا خواهد بود. بردار ویژه های باقیمانده به مقادیر ویژه ی صفر مرتبط می شوند.

به بردارهای ویژه ی V_i ی $A^T A$ 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 $A V_i$ بردار های ویژه ی $C = A A^T$ هستند.

در ادامه ی 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 ی V_i ی L را پیدا می کنیم. این بردار ها ، ترکیب های خطی از M تصویر چهره ی مجموعه ی 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 ی تعداد پیکسل ها در تصاویر (N^2) به مرتبه ی تعداد تصاویر در مجموعه ی تمرین (M) کاهش می یابند. در عمل ، مجموعه ی تمرین تصاویر چهره به نسبت کوچک خواهند بود ($M \ll N^2$) و محاسبات کاملاً قابل مدیریت می شود. مقادیر ویژه ی مرتبط ، به ما این قابلیت را می دهند تا بردار های ویژه را بر اساس فایده ی آنها در مشخص سازی اختلاف میان تصاویر رتبه بندی کنیم.

فصل سوم :

تشخیص چهره با استفاده از

HMM ۱

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 ایستا 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 مواقع فریم ها^۱ 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 multivariate، $b_j(O_t)$ 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 پارامترهای 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.

$$B = \{b_j(\cdot) : 1 \leq j \leq N\}$$

که تابع 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 $Q = q_1 \dots q_T$ و دنباله ی مشاهدات مربوطه $O = o_1 \dots o_T$ به وسیله ی ضرب هر احتمال انتقال در هر احتمال خروج در هر مرحله ی 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 ی 3.1 قابل محاسبه نیست. به هر حال احتمال $P(O | \lambda)$ به وسیله ی جمع کردن همه ی دنباله حالت های ممکن قابل محاسبه است.

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

جاذبه ی کلیدی HMM ها آن است که یک رویه ی ساده برای پیدا کردن پارامترهای λ که معادله ی 3.2 را ماکزیمم می کنند ، وجود دارد. این رویه معمولاً به عنوان تخمین مجدد 'Baum-Welch' معرفی شده به [11] Baum ارجاع می شود و برای عملیاتش به الگوریتم رو به جلو - رو به عقب نیاز دارد که در آن احتمال رو به جلو^۱ $P(o_1 \dots o_t, q_t = j | \lambda)$ و احتمال رو به عقب $P(o_{t+1} \dots o_T | q_t = j, \lambda)$ به طور کارایی به وسیله ی یک تکرار ساده تعریف می شود. متغیرهای 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

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

$$1 \leq j \leq N \quad (3.7)$$

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

$$1 \leq i \leq N \quad (3.8)$$

3. در پایان

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

محاسبه ی مستقیم $P(\mathbf{O} | \lambda)$ بر طبق معادله ی 3.2 از مرتبه ی محاسباتی TN^2 است. در حالیکه محاسبات از طریق الگوریتم رو به جلو - رو به عقب معادله ی 3.9 از مرتبه ی 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 ی t ی دنباله ی مشاهده ی داده شده ی O را به دست می دهد. این متغیر به صورت $\gamma_t(j) = (q_t = j | O, \chi)$ 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 ی جدید از پارامتر های HMM برای هر حالت j ، به وسیله ی محاسبه ی میانگین های وزن داده شده ، پیدا می شود. مدل توصیف شده به وسیله ی مجموعه ی جدید پارامتر ها به صورت $\bar{\chi} = \{\bar{\Pi}, \bar{A}, \bar{B}\}$ 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 ی مشاهده ی 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 :

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

با استفاده از مفهوم رخداد های شمارش¹ ، پارامتر های مدل برای $\bar{\chi}$ می تواند به صورت زیر تخمین مجدد زده شود :

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

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

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

$$\bar{\mathbf{V}}_i = \frac{\sum_{t=1}^T \gamma_t(i) \cdot (\mathbf{o}_t - \bar{\mu}_i)(\mathbf{o}_t - \bar{\mu}_i)^T}{\sum_{t=1}^T \gamma_t(i)} \quad (3.16)$$

که علامت پریم بر ترانهاده ی بردار اشاره دارد و $\bar{\mu}_i$ و $\bar{\mathbf{V}}_i$ تخمین های میانگین و ماتریس کواریانس تابع 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 ماکزیمم ساختن مجموعه ای از دنباله های مشاهده ، به وسیله ی جمع کردن numerators و denominators 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

¹ counting occurrences

کند. این احتمال بایستی دقیقاً احتمال کل تعریف شده در معادله ی 3.2 باشد. به هر حال، در عمل آسان تر آن است که دنباله ی حالتی که معادله ی 3.1 را ماکزیمم می کند پیدا کنیم و در عوض احتمال ماکزیمم 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 مشاهده ی t محاسبه می کند و در حالت j پایان می پذیرد، به وسیله ی کمیت $\delta_t(j)$ به صورت زیر تعریف می شود :

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

یک رویه ی ابتدایی که برای محاسبه ی مقادیر $\delta_t(j)$ قابل استفاده است به صورت زیر است :

مقدار دهی اولیه

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

$$1 \leq j \leq N$$

در ادامه

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

$$1 \leq t \leq T-1,$$

$$1 \leq j \leq N$$

در پایان

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

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

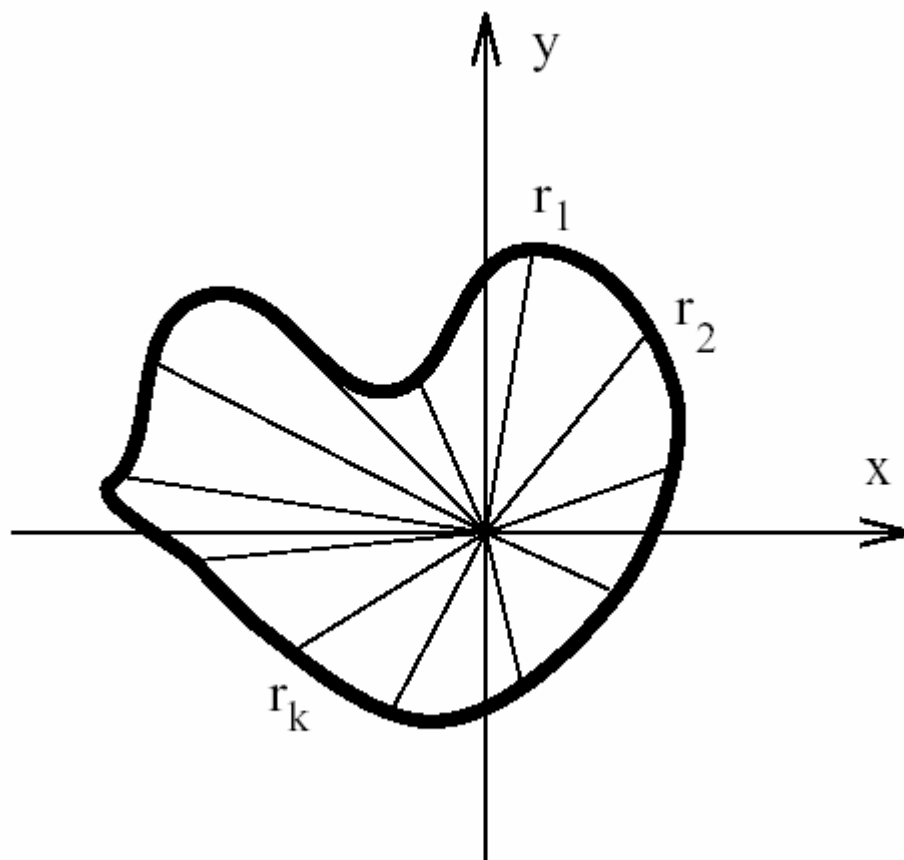
$$q_t = \psi_{t+1}(q_{t+1})$$

HMM ها در بینایی

در بیشتر موارد HMM ها در کاربرد های 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 قبلا گفته شد ، HMM ها ویژگی های آماری دنباله های مشاهده ی یک بعدی را مدل می کنند و داده ی گفتاری به طور طبیعی بر روی محور زمان یک بعدی است.

[12] He and Kundu ، 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 ثقل¹ 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 100٪ گزارش شد.

¹ gravity



شکل 1-3. تکنیک نمونه برداری شعاعی

کاربرد دیگر HMM های گسسته به وسیله ی [13] kundu 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 گسسته معرفی شد.

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

تشریح یک معماری

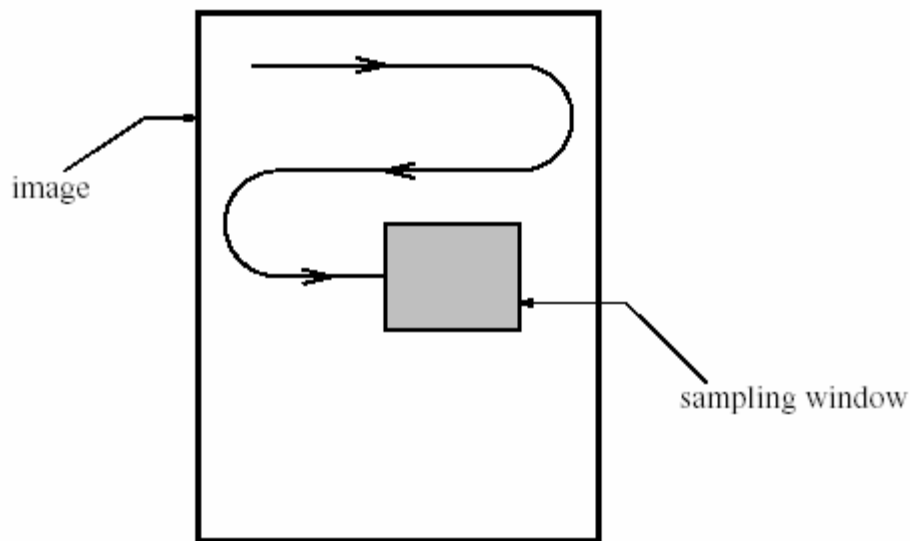
در ادامه ، مسئله ی تشخیص چهره از پرسپکتیو تشخیص الگوی آماری بررسی می شود. به طور معمول ، یک چهره قابل تقسیم به تعدادی ناحیه مانند دهان ، چشم ، بینی و ... است و اگر مکان این نواحی به طور مطمئنی قابل بدست آوردن بود ، آنگاه تکنیک های استاندارد مطابقت الگو ، به طور مجزا برای هر ناحیه به جهت محاسبه ی یک متریک فاصله ی کلی ، قابل استفاده هستند. به هر حال ، در عمل مکان یابی نواحی کار بسیار دشواری است.

یک راه حل ممکن برای مسئله ای که در بالا ذکر شد ، آن است که بین نواحی چهره و حالت های HMM چگالی پیوسته ، ارتباط برقرار کنیم. در پی این کار ، این امکان به وجود می آید تا محدوده های بین نواحی به وسیله ی انتقال های احتمالی بین حالت ها ارائه شوند و تصویر واقعی در یک ناحیه به وسیله ی یک توزیع گوسی multivariate مدل شود. در کل ، HMM نیاز به دو بعدی بودن دارد. به هر حال ، در ادامه ، به یک تقریب مرتبه اول¹ در شرایطی که نواحی چهره ، محدود به باندهای افقی هستند ، دست می زنیم.

در اینجا ، یک مدل با استفاده از چندین تصویر چهره برای یک موضوع یکسان تمرین داده می شود. هر تصویر یک دنباله ی مشاهده $O = O_1 \dots O_T$ تولید می کند. مشاهده ی O_t از یک بلاک شامل پیکسل ها در تصویر دو بعدی به وسیله ی میانگین های یک پنجره ی نمونه برداری که تصویر را به ترتیبی که در شکل 3.2 نشان داده اسکن می کند ، به

¹ first – order approximation

دست می آید. پیکسل های پنجره ی نمونه برداری شامل مقادیر سطح شدت ، در بردار ستونی O_t مرتب می شوند. فرآیند نمونه برداری مقداری که یک مدل می تواند موفق باشد را تعیین می کند. HMM ها توزیع آماری همه ی دنباله های مشاهده ی ارائه شده ی مرتبط با یک کلاس خاص یا در این زمینه مرتبط با یک موضوع خاص را ارائه می کنند. در مورد تصاویر چهره ، تعدادی از دید ها و حالات¹ مختلف هر چهره قابل ترکیب در یک مدل آماری منفرد است.



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

در ادامه دو توپولوژی پایه ای HMM بررسی می شوند : **ergodic** و بالا - پایین². توپولوژی **ergodic** مدلی ساده و بدون قید از داده ها تولید می کند. این مدل هیچ استفاده ای از اطلاعات ساختاری نمی کند.

¹ expressions

² top - bottom

رویه های تمرین و تست

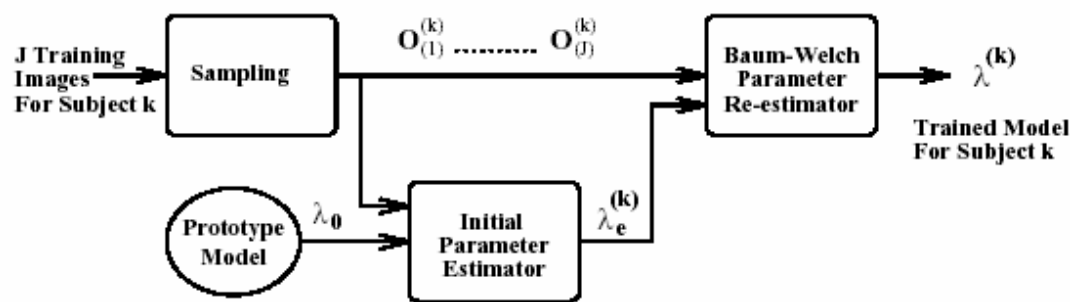
فرآیند تمرین دادن برای هر موضوع S در پایگاه داده شامل مراحل زیر است که در دیاگرام شکل 3-3 نیز خلاصه شده است :

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

2. مدل HMM نمونه λ_0 با هدف تعیین تعداد حالت های HMM ، انتقال های مجاز بین حالت ها و اندازه ی بردار های دنباله ی مشاهده ساخته می شود.

3. مجموعه ای از مقادیر پارامتر های اولیه با استفاده از داده های 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 دور های بعدی ، بخش بندی یکنواخت با حاشیه بندی Viterbi جایگزین می شود. نتیجه یک HMM اولیه تخمین $\chi_e^{(k)}$ که به عنوان ورودی به مرحله ی تخمین مجدد استفاده می شود.

4. پارامتر های HMM با استفاده از روش Baum-Welch مجدداً تخمین زده می شوند. پارامتر های مدل برای ماکزیمم سازی محلی احتمال مشاهده ی داده ی تمرینی برای هر مدل تنظیم می شوند. نتیجه ی این فرآیند یک HMM به نام $\chi^{(k)}$ است که برای ارائه ی موضوع k در پایگاه داده استفاده می شود.



شکل 3-3. دیاگرام بلوکی تکنیک تمرین دادن

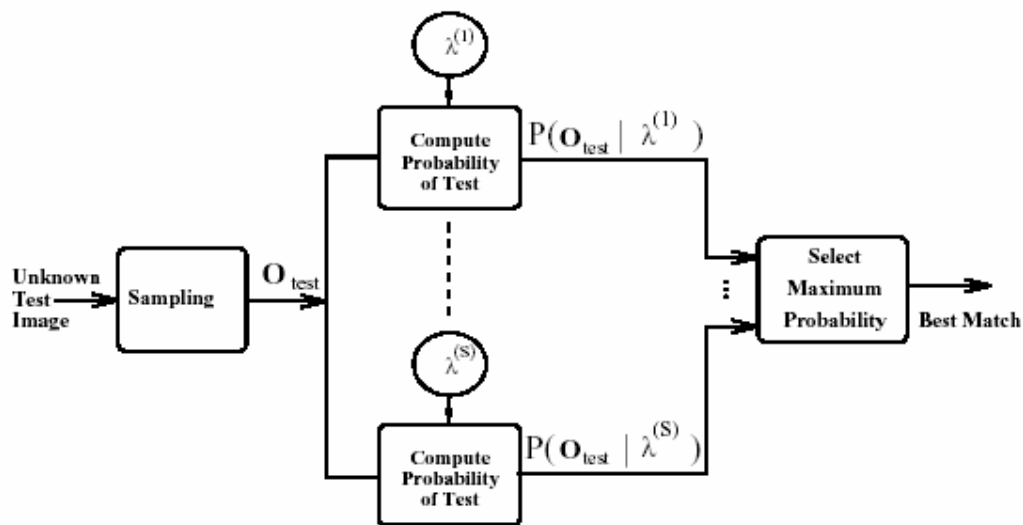
تشخیص به وسیله ی یک تشخیص دهنده ی Viterbi انجام می شود. تصویر تست با هر یک از HMM های ارائه دهنده ی موضوعات مختلف مطابقت داده می شود و بیشترین مطابقت انتخاب می شود. روند تشخیص مراحل زیر را شامل می شود که در دیاگرام شکل 3-4 خلاصه شده است :

1. تصویر تست نا شناخته برای تولید دنباله ی مشاهده O test نمونه برداری می شود.
2. دنباله ی مشاهده با هر مدل چهره به وسیله ی محاسبه ی احتمال مدل مطابقت داده می شود :

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

در عمل همانطوری که در بخش قبلا شرح داده شد ، مقدار احتمال ماکزیمم استفاده می شود.

3. مدلی با بیشترین احتمال انتخاب می شود و این مدل هویت چهره ی ناشناس را معرفی می کند.



شکل 4-3. دیاگرام بلوکی تشخیص دهنده ی چهره

توپولوژی HMM

یک HMM دو بعدی کاملاً مرتبط ، برای مدل سازی یک تصویر دو بعدی مطلوبست. به هر حال پیچیدگی محاسباتی برای یک شبکه ی دو بعدی کاملاً مرتبط ، همانطور که در Levin و Pieraccini [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.

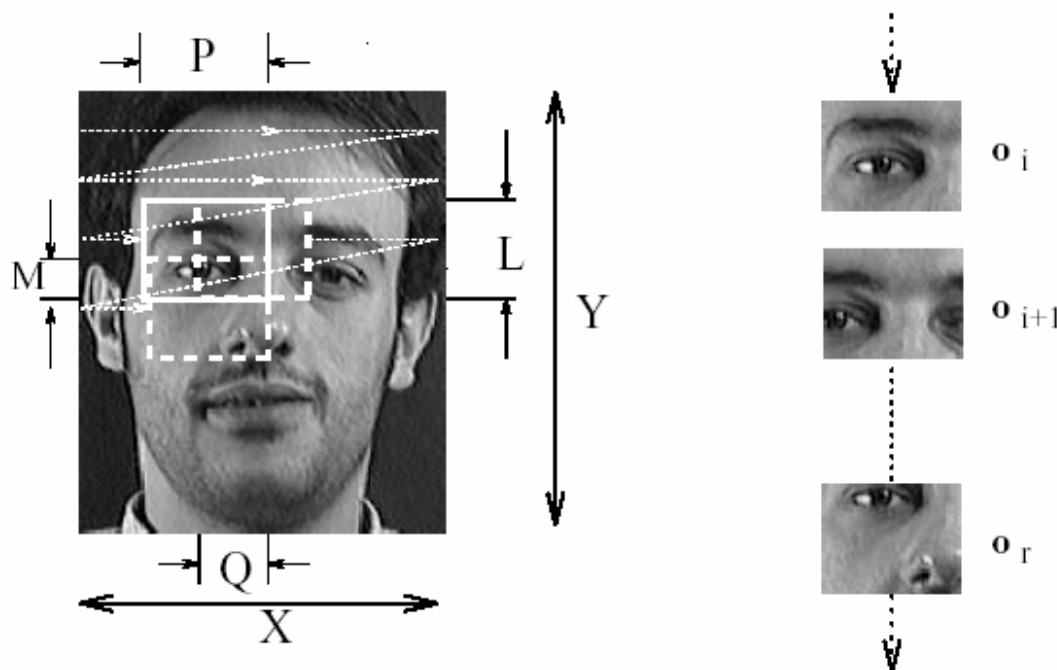
تکنیک های نمونه برداری که در ادامه معرفی می شوند ، هدف استخراج یک دنباله ی مشاهده ی یک بعدی از تصویر دو بعدی را دنبال می کنند. دنباله ی مشاهده به وسیله ی استخراج بلوک های پیکسل از یک تصویر با استفاده از پنجره ی نمونه برداری ساخته می شود. هر مشاهده یک بردار ستونی شامل سطح شدت پیکسل ها در پنجره است. دنباله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

وسیله ی اسکن کردن با یک ترتیب تشکیل می شود. روشی که پنجره ی نمونه برداری تصویر را اسکن می کند بر توپولوژی مدل که داده ها را توصیف می کند ، تاثیر گذار است. در بخش های زیر ، دو تکنیک نمونه برداری مختلف آنالیز می شوند. در مورد اول ، پنجره ی نمونه برداری مستطیلی تصویر را از چپ به راست و از بالا به پایین اسکن می کند. که یک مدل **ergodic** برای ارائه ی دنباله ها استفاده می شود. در مورد دوم ، پنجره ی نمونه برداری بلوک هایی از خطها را از بالا به پایین چهره استخراج می کند. با استفاده از این مزیت موجود که ویژگی ها (چشم ها ، بینی ، دهان و ...) با یک ترتیب قابل پیش بینی ظاهر می شوند ، یک **HMM** غیر **ergodic** (بالا - پایین) برای ارائه ی تصویر چهره ساخته می شود. مدل **ergodic** از اطلاعات ساختاری هیچ استفاده ای نمی کند. از طرف دیگر رویکرد بالا - پایین ، از الگوهای داده ای تا حدودی بهره می برد.

Ergodic HMMs

در مدل های **ergodic** ، هر حالت ، از هر حالت دیگر قابل دسترسی است. این بدان معناست که همه ی ضرایب ماتریس انتقال **A** مثبت هستند. به طور کلی مدل های **ergodic** زمانی که قید های محدود برای سیگنال می توانند بکار روند ، استفاده می شوند و کلی ترین نوع **HMM** ها هستند. برای تشریح نحوه ی کار آن ها ، یک **ergodic HMM** برای یک تصویر نمونه برداری شده با استفاده از شکل 3-5 ساخته می شود که یک پنجره ی **P*L** تصویر را از چپ به راست و از بالا به پایین اسکن می کند. همانطوری که پنجره ی نمونه برداری از چپ به راست روی یک خط حرکت می کند ، هر مشاهده تعداد **Q** ستون همپوش با مشاهده ی قبل از خود دارد. زمانی که پنجره به پایان خط کنونی رسید ، پنجره ی نمونه برداری به شروع خط باز می گردد و آن قدر به سمت پایین شیفست می کند تا به میزان **M** خط با پنجره ی بالایی خط قبل هم پوشانی

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



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

شکل 3-6 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 ی 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 3-6 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 ی 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 دهند.

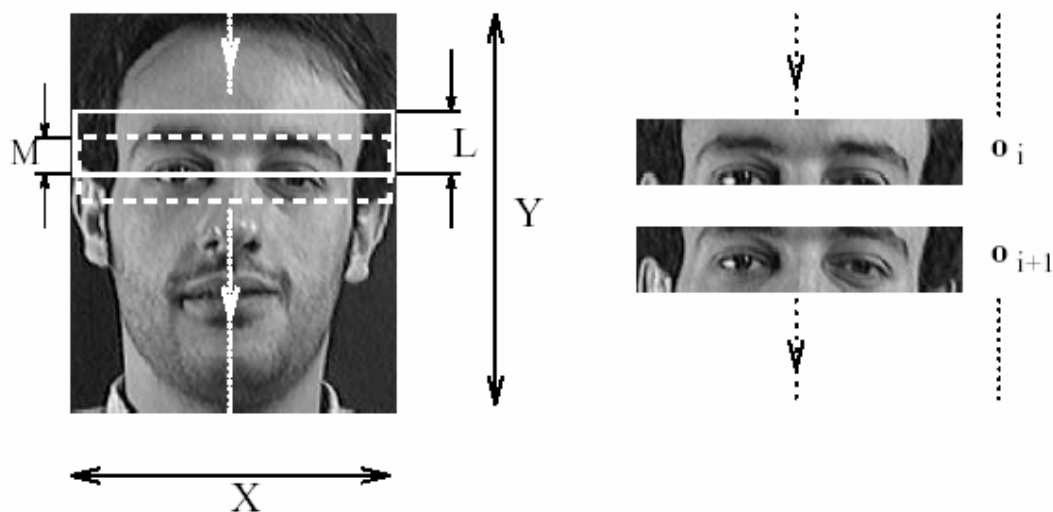


شکل 3-6. داده تمرینی و میانگین های مدل برای ergodic HMM

HMM های بالا - پایین

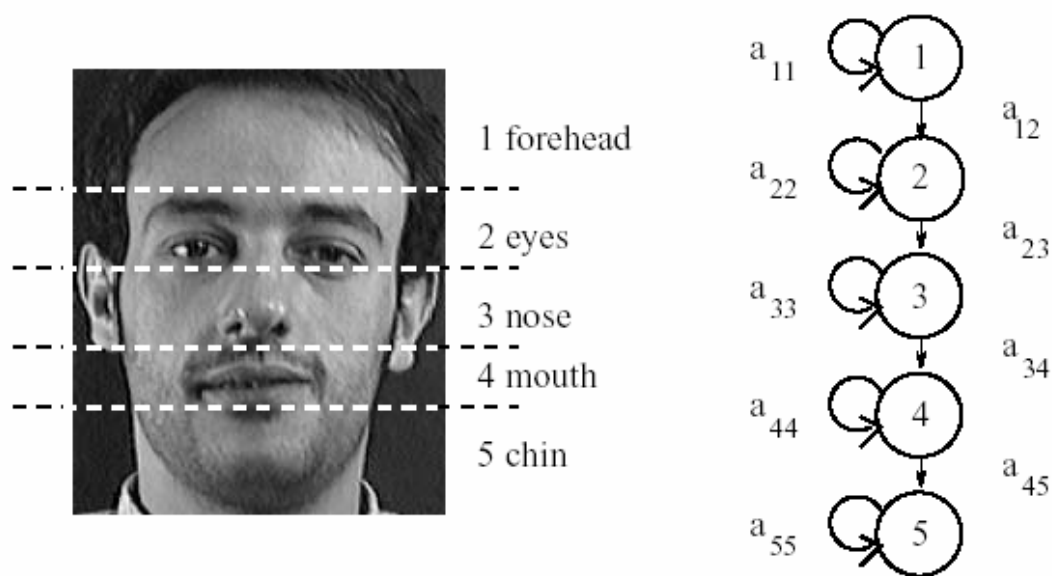
حالت های یک HMM در مدل ergodic به طور دلخواه می توانند ارتباط داده شوند که جزئیاتی از آن را در بخش قبل ملاحظه فرمودید. به هر حال ، برای کاربرد های تشخیص الگو ، بهتر است قید هایی بر روی انتقال حالت های مجاز گذاشته شود تا ویژگی های داده را reflect کند. به خصوص ، توپولوژی های HMM به نام چپ - راست¹ اغلب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

¹ Left - right



شکل 3-7. 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 ی 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 ی 1960 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 ی دسته بندی دو کلاسی ، هدف جداسازی دو کلاس بوسیله ی 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 ی بین صفحه و نزدیکترین نقطه ی داده ای هر کلاس به صفحه) را افزایش می دهد. این دسته بند خطی ، صفحه ی جداکننده ی بهینه¹ نامیده می شود. به مسئله ی جداسازی مجموعه ی بردارهای تمرین متعلق به دو کلاس مجزای

$$(x_1, y_1), \dots, (x_l, y_l), \text{ where } x_i \in R^m, \\ y_i \in \{-1, +1\}$$

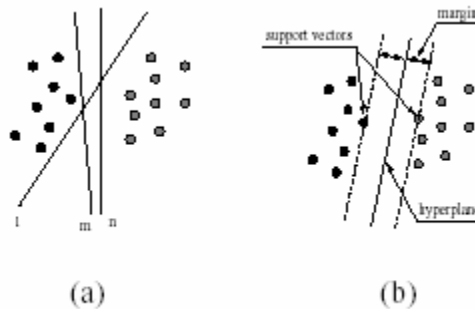
با صفحه ی

$$wx + b = 0.$$

¹ Optimal separating hyperplane (OSH)

توجه کنید . به مجموعه ی بردارهایی مجزا شده ی بهینه¹ به وسیله ی صفحه گفته می شود ، اگر بدون خطا جدا شوند و حاشیه ماکزیمم شود. یک صفحه ی 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 4-1. دسته بندی دو کلاس به وسیله ی صفحه ها

(a) صفحه های دلخواه l, m, n

(b) صفحه ی جداساز بهینه با بیشترین حاشیه

یک صفحه ی 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 ی نقطه ی \mathbf{x} از صفحه این گونه محاسبه می شود :

¹ Optimally separated

$$d(\mathbf{w}, b; \mathbf{x}) = \frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|} \quad (4-2)$$

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 :

$$\Phi(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 \quad (4-3)$$

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

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i \{y_i [(\mathbf{w} \cdot \mathbf{x}_i) + b] - 1\} \quad (4-4)$$

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

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left\{ \min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha) \right\} \quad (4-5)$$

راه حل برای مسئله ی 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 معادله ی 4-6 با قیود (4-7 و 4-8) ، ضرایب لاگرانژ را حاصل می شود و OSH این گونه ارائه می شود

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

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

که \mathbf{x}_r , \mathbf{x}_s بردار های پشتیبانی هستند که قیود زیر را رعایت می کنند

$$\bar{x}_1, \bar{x}_2 > 0, \quad y_1 = 1, \quad y_2 = -1$$

برای نقطه ی داده ای جدید X دسته بندی این گونه است :

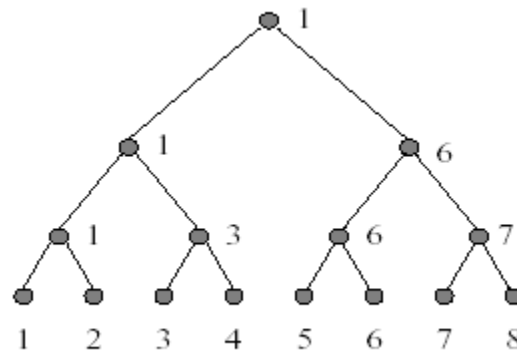
$$f(\mathbf{x}) = \text{sign}(\bar{\mathbf{w}} \cdot \mathbf{x} + \bar{b})$$

که البته این دسته بندی خطی بود.

تشخیص چند کلاسی

در بخش قبلی پایه های تئوری ماشین های بردار پشتیبان برای دسته بندی دو کلاس تشریح شد. یک تشخیص الگوی چند کلاسی می تواند به وسیله ی ترکیب ماشین های 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 دسته بندی بین هر کلاس و کلاس های باقیمانده است. دیگری استراتژی « یک در مقابل یک » برای دسته بندی هر جفت است. در شرایطی که دسته بندی اول به دسته بندی مبهم منجر می شود. دومی را برای سیستم تشخیص چهره وفق می دهیم. پیشنهاد می کنیم که یک درخت باینری پایین به بالا برای دسته بندی ساخته شود. فرض کنید هشت کلاس در مجموعه ی داده وجود دارد. درخت تصمیم در شکل 2-4 نشان داده شده است، جاییکه اعداد 1 تا 8 نشانگر هشت کلاس ها هستند. توجه کنید که در استفاده از این اعداد هیچ ترتیبی بکار نرفته است. برای یک چهره ی مورد تست ، مقایسه ای بین هر جفت کلاس صورت می گیرد و یک کلاس به عنوان برنده ی مقایسه ی دو کلاس انتخاب می شود. کلاس های 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 نقطه ی درخت ظاهر شود.



شکل 2-4. ساختار درخت دودویی برای 8 کلاس تشخیص چهره. برای چهره ی تست ورودی ، آن چهره با هر دو جفت مقایسه می شود و برنده ها 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 ایده ی مهندسی معکوس مطرح می شود. چرا بایستی چیز هایی را که هم اکنون وجود دارند ، اختراع کنیم. در میانه های قرن بیستم ، ایده ای برای شبیه سازی مغز انسان از طریق طراحی پایین به بالای سیستم با شروع از نرون¹ های به هم متصل و رسیدن به شبکه هایی که اعمال اصلی سیستم عصبی انسان را شبیه سازی می کنند ، مطرح شد.

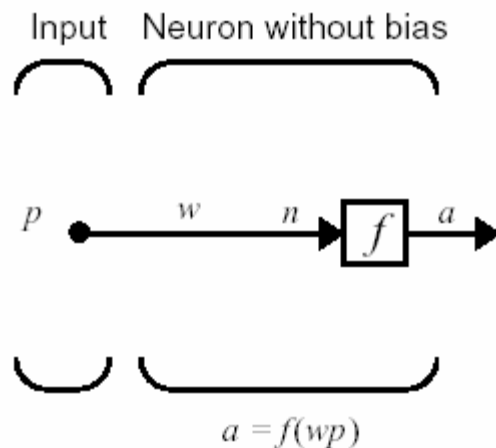
با گذر زمان ، شبکه های عصبی² خودشان را به عنوان مدل محاسباتی مورد استفاده مطرح کردند. شبکه های عصبی معمولاً در انجام وظایف کامپیوتر های عام (عملیات با اعداد ، اعمال منطقی و ...) نادقیق هستند ، اما در زمینه هایی که کامپیوتر ها در آنجا به مشکل بر می خورند ، کارآیی بیشتری دارند مانند زمینه هایی که الگوریتم 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.

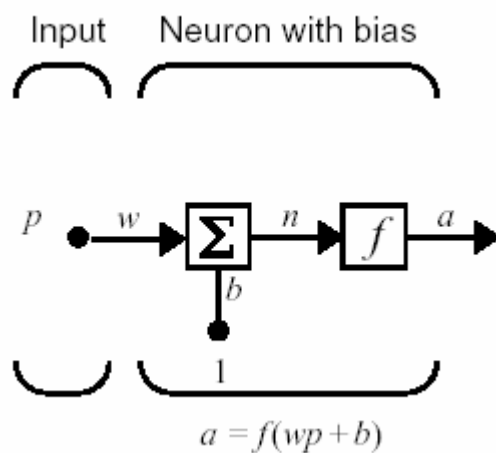
¹ neuron

² neural network



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

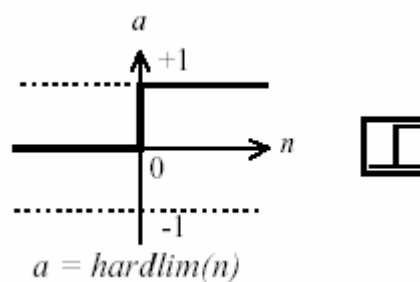
ورودی اسکالر p از طریق ارتباطی که مقدار آن را در وزن اسکالر w ضرب می کند به صورت حاصلضرب wp که اسکالر است ، منتقل می شود. در این مرحله ، ورودی وزن داده شده ی wp بخه عنوان آرگومان به تابع f منتقل می شود و این تابع خروجی a را تولید می کند. نرون شکل 5-2 یک مقدار بایاس b دارد. این مقدار به حاصلضرب wp اضافه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



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

توابع انتقال^۱

سه تابع انتقال که معمولاً مورد استفاده قرار می گیرند در شکل 5-3 ، 5-4 و 5-5 نشان داده شده است.

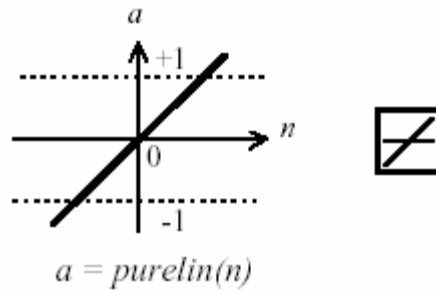


شکل 5-3. تابع انتقال hard-limit

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

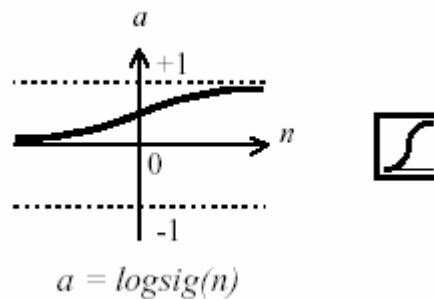
تابع انتقال خطی در شکل 5-4 نشان داده شده است. نرون هایی با این تابع انتقال در تقریب زننده های خطی استفاده می شوند.

¹ Transfer function



شکل 5-4. تابع انتقال خطی

تابع سیگموئید در شکل 5-5 نشان داده شده است. مقدار خروجی این تابع در بازه ی بین مقادیر صفر و یک محصور است. این تابع انتقال معمولاً در شبکه های انتشار عقب¹ 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 5-6 نشان داده شده است. بردار ورودی

p

$$p_1, p_2, \dots, p_R$$

در بردار وزن W

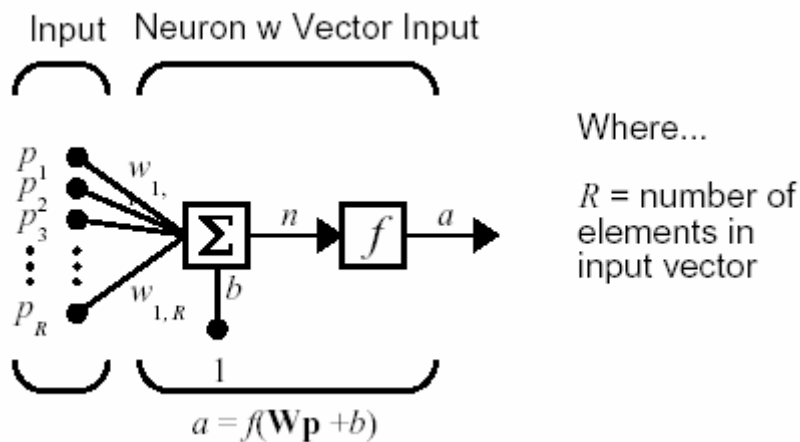
¹ backpropagation

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

به روش ضرب اسکالر ، ضرب می شود.

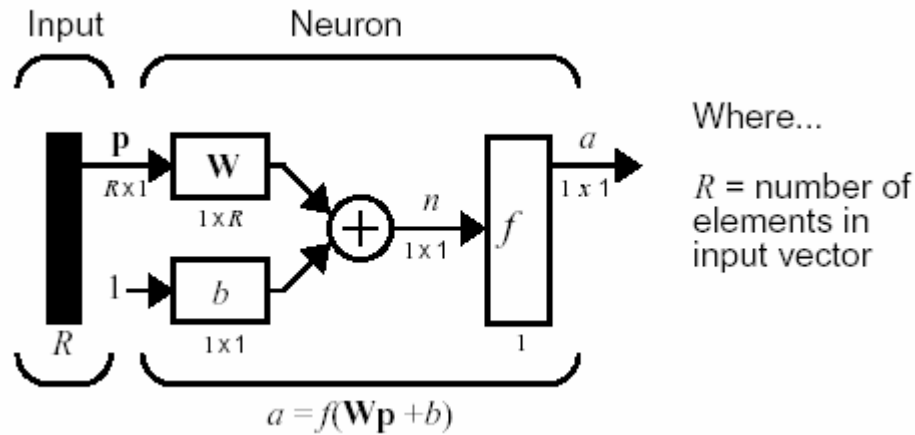
$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b$$

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



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

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



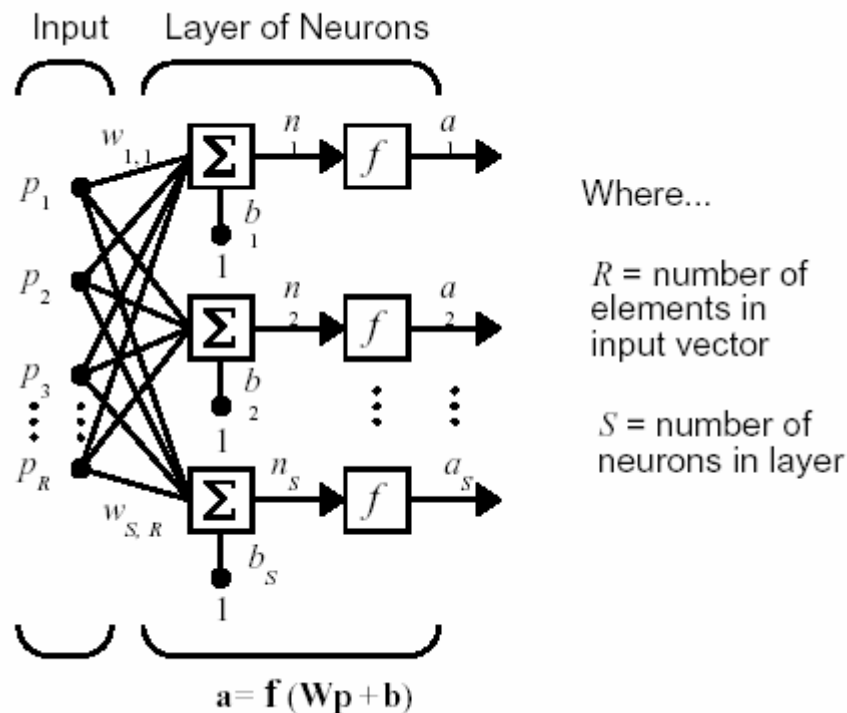
شکل 5-7. یک نرون با بردار ورودی R عنصری (نشانه گذاری سطح بالاتر)

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

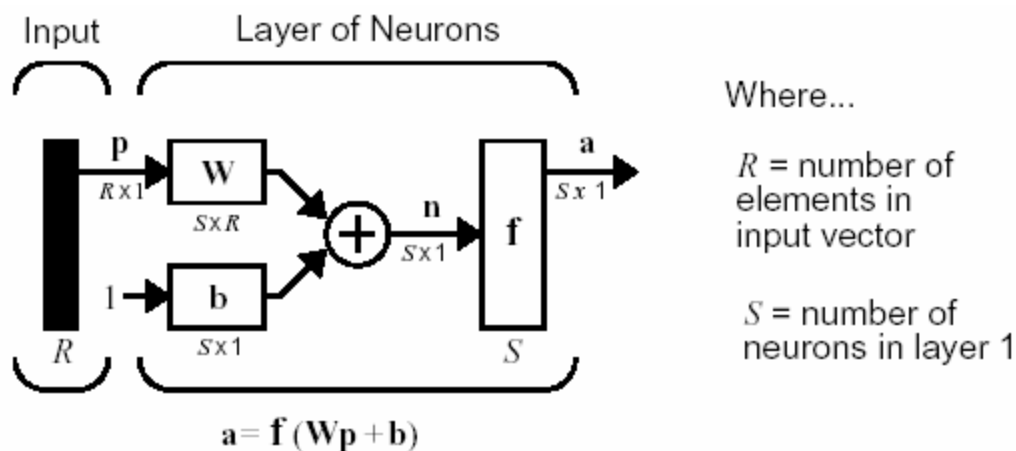


شکل 5-8. یک شبکه ی یک لایه ای با R عنصر ورودی و S نرون

در این شبکه ، هر عنصر بردار ورودی p به هر ورودی نرون از طریق ماتریس وزن W مرتبط می شود. نرون i ام جمع کننده ای دارد که ورودی های وزن داده شده را با مقدار بایاس برای تولید مقدار خروجی $n(i)$ ، جمع می کند . $n(i)$ های بدست آمده از جمع کننده ها به عنوان ورودی به توابع انتقال فرستاده می شوند و در نهایت مقادیر a_i به عنوان خروجی لایه محسوب می شوند و بردار a را می سازند. الزامی وجود ندارد که تعداد ورودی ها به یک لایه با تعداد نرون های آن لایه برابر باشد. وزن های 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. شبکه ی تک لایه ی مورد بحث با نشانه گذاری کوتاه تری در شکل 5-9 نشان داده شده است.

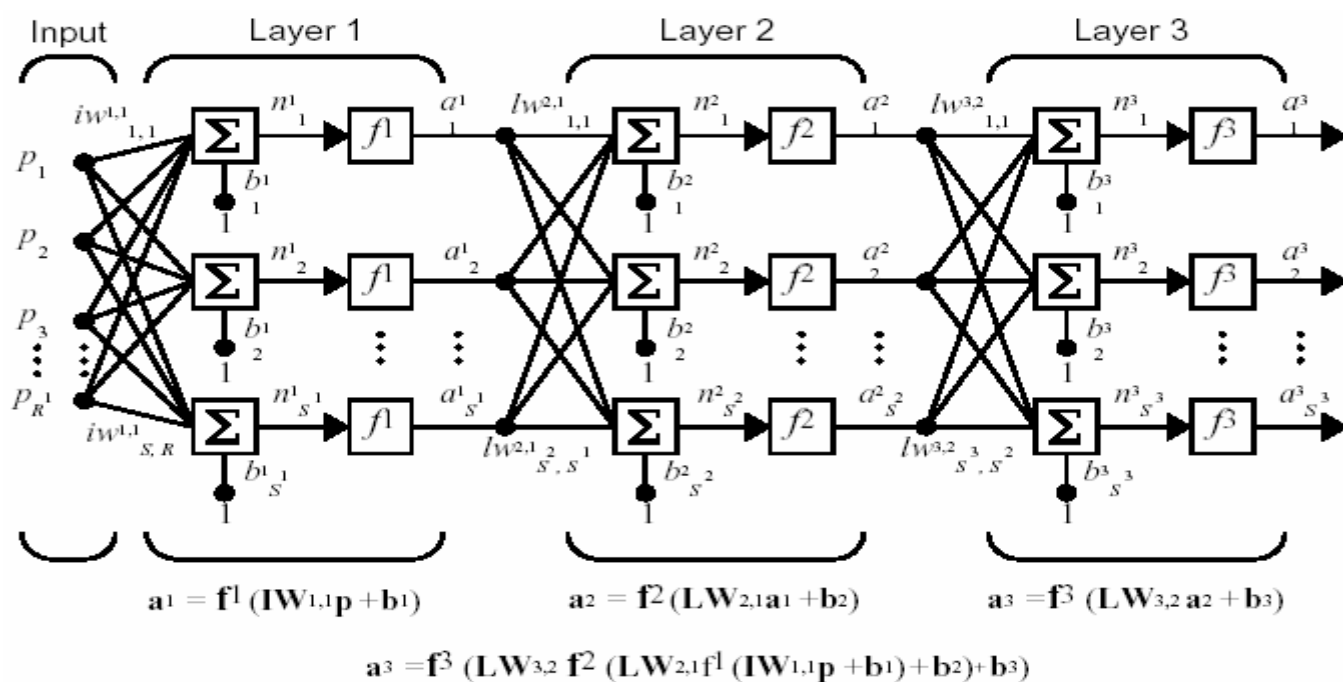


شکل 5-9. یک شبکه ی تک لایه ای با R عنصر ورودی و S نرون (با نشانه گذاری سطح بالاتر)

چندین لایه از نرون ها

یک شبکه می تواند 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 ماتریس های وزن، بردار های خروجی و ...، برای هر یک از لایه ها در شکل ها، شماره ی لایه ی مربوطه را به متغیر

های مورد نظر اضافه می کنیم. می توان استفاده از این نشانه ی لایه را در شبکه ی سه لایه ای که در شکل 5-10 نشان داده شده است ، ملاحظه نمود.



شکل 5-10. شبکه ی سه لایه ای

یادگیری شبکه های عصبی

شبکه های عصبی قادر به یادگیری از نمونه های ارائه شده است. در شبکه های عصبی، فرآیند یادگیری ، تنظیم وزن های ارتباطات بین نرون هاست. 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 مرحله ی یادگیری ، برای هر نمونه ی یادگیری هر دوی ورودی و خروجی مورد نظر آن ورودی را داریم. بنابراین ، تعدادی جفت های تمرینی داریم (بردار های ورودی و خروجی ممکن است به ترتیب به عنوان « سوال » و « پاسخ صحیح » دیده شوند.) . این جفت ها به طور تکراری از شبکه گذرانده می شوند و در هر تکرار الگوریتم های یادگیری به کار برده می شوند و وزن ها تنظیم می شوند تا شبکه بتواند به ورودی ها ، پاسخ صحیح دهد (مثال نوعی این کلاس یادگیری ، الگوریتم « انتشار به عقب خطا » 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 فاصله ی (مثل فاصله ی همینگ) یک بردار ورودی با بردارهای ورودی دیگر انجام می دهد. به هر حال نتایج این نوع یادگیری بایستی آنالیز شود چرا که دسته بندی ممکن است نادرست باشد.

اکنون برای نمونه ، تعریفی رسمی از الگوریتم انتشار به عقب (برای یک شبکه ی سه لایه) ارائه می شود.

1. بردار ورودی به شبکه ارائه می شود.

2. feedforward انجام می شود 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 نرون های لایه ی قبلی محاسبه کند.

3. خطا در لایه ی 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. خطا به لایه ی میانی انتشار عقب می شود

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

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

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

و

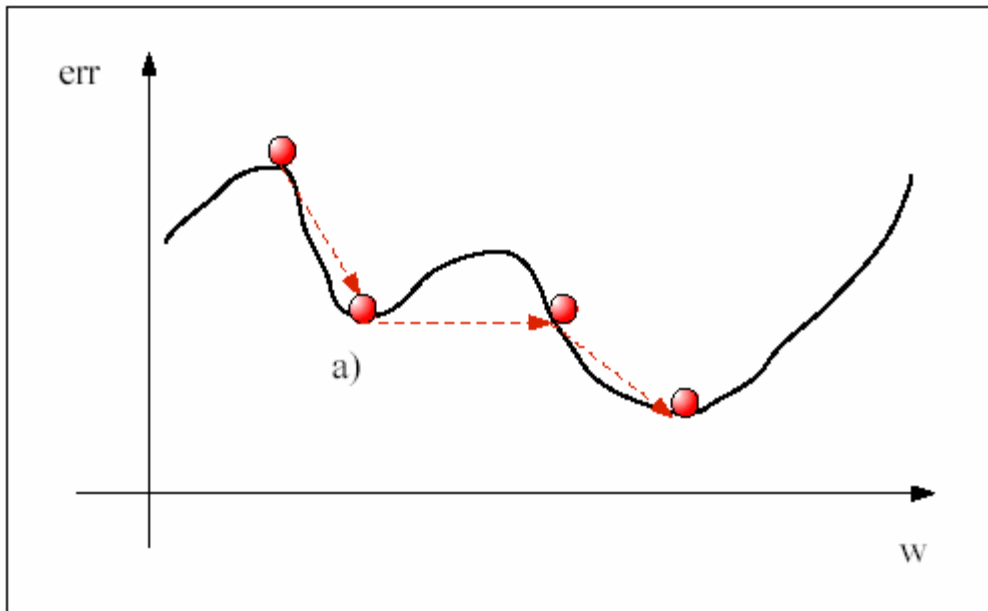
$$w_{ij}^h(t+1) = w_{ij}^h(t) + \alpha err_j^h x_i$$

می توانند تنظیم شوند که α ضریب یادگیری و x_i نرون i ام در لایه ی 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 خطا را 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 ی تمرین، تمام محدوده ی 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 این دانش را تعمیم دهد. این دلیلی است که 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 ی شکل های ممکن اعداد نیست و ممکن است مجموعه ی تمرینی متشکل از تعدادی از نمونه های متداول 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 ی 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 . همانطور که در فصل های قبل نیز گفته شده ، در فرآیند دسته بندی ، کلاسی که یک الگوی ورودی عضو آن است مشخص می گردد. در این جا به نحوه ی سامان دهی شبکه ی عصبی و جفت های تمرین برای دسته بندی الگو ها می پردازیم.

اگر جفت های 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. نکته ای که می ماند آنست که خروجی ها را چگونه طراحی کنیم تا دسته بندی را بر آورده 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 5-12 این کد گذاری را برای تصاویر چهره ی ورودی به شبکه ملاحظه می کنید.



class1 : 1000



class2 : 0100



class3 : 0010



class4 : 0001

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 چهره ی دو بعدی به جهت ارائه به شبکه ، پیمایش سطر به سطر یا ستون به ستون تصویر باشد و از کنار هم قرار دادن سطر ها یا ستون ها ، یک بردار به دست آید. در تشخیص چهره ، این بردار ورودی که به شبکه عصبی ارائه می شود ، همان بردار ویژگی است. ویژگی هایی که در مراحل قبل فرآیند تشخیص چهره از تصویر استخراج شده اند و برای دسته بندی به شبکه ی عصبی ارائه می شوند . شبکه ی عصبی همانطور که قبلا بحث شد ، فرآیند یادگیری را پشت سر می گذرانند تا برای دسته بندی چهره ها 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 wrapping 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 " , Proceodings 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