



وزارت علوم تحقیقات و فناوری  
موسسه آموزش عالی سلمان

## پایان نامه کارشناسی

### پردازش تصویر

ارائه دهنده: مهدی ارباب تفتی  
استاد دفاع: مهندس معارف دوست  
استاد راهنما: مهندس حسن شاکری

دیماه ۱۳۸۷

## چکیده

OCR یا Optical Character Recognition به معنای

شناسایی حروف با کمک ابزار نوری است و یک شاخه از علم پردازش سیگنال را تشکیل می دهد . OCR شامل تصویر برداری از روی کاغذ و استخراج کلمات از آن است.

متن به صورت مستقیم در تصویر قابل دسترسی نیست زیرا حروف در یک متن اسکن شده از مجموعه ای از نقاط تشکیل می شود که در کنارهم قرار گرفتن این نقاط مطابق الگوهای معین، تصویر حروف مختلف را می سازد

## فهرست

فصل ۱: مقدمه.....	۸
۱-۱: آشنایی با OCR.....	۹
۱-۲: پردازش تصاویر ماهواره.....	۱۲
۱-۳: پردازش تصویر در صنعت و پزشکی.....	۱۲
فصل ۲: مراحل OCR.....	۱۵
۲-۱: سیستمهای بازشناسی حروف.....	۱۶
۲-۲: اسکن کردن صفحات.....	۲۱
۲-۳: استخراج نواحی شامل متن.....	۲۳
۲-۴: شناسایی حروف.....	۲۷
۲-۵: ارائه نتایج.....	۳۰
فصل ۳: شبکه های عصبی.....	۳۱

۳-۱: معرفی شبکه های عصبی.....	۳۲
۳-۱-۱: انواع شبکه های عصبی.....	۳۳
۳-۲: نرون.....	۳۷
۳-۳: شبکه های Multilayer Perceptron.....	۳۹
۳-۴: Overlearning.....	۴۳
فصل ۴: شرح یک پروژه انجام شده.....	۴۸
۴-۱: شرح پروژه.....	۴۹
۴-۲: تصاویر مورد استفاده.....	۴۹
۴-۳: تبدیل تصاویر به بردارهای ویژگی.....	۵۰
۴-۴: آموزش شبکه عصبی.....	۵۴
۴-۵: طراحی ابتدایی.....	۵۵
۴-۶: تغییرات و ساده سازیها.....	۵۹
فصل ۵: ویژگی های متون فارسی.....	۶۲
۵-۱: خصوصیات متون فارسی.....	۶۳

فصل ۶: بررسی چند محصول OCR	۶۶
۶-۱: چند نمونه OCR	۶۷
۶-۲: آموزش Readiris Pro 11	۶۷
۶-۳: آموزش VajehShenas V.1	۷۱
۶-۴: مقایسه VajehShenas با Readiris	۷۵
فصل ۷: نتایج و پیشنهادات	۷۷
مراجع	۸۱

## فهرست شکل ها

شکل ۲-۱ نمونه فرم اسکن شده.....	۲۵
شکل ۲-۲ یک نرون با تابع فعال ساز.....	۳۷
شکل ۲-۳ چند تاج فعال سازی رایج.....	۳۹
شکل ۲-۴ Multilayer Perceptron .....	۴۱
شکل ۲-۵ overfitting چند جمله ای.....	۴۵
شکل ۲-۶ overlearning در شبکه عصبی.....	۴۶
شکل ۳-۱ یافتن Bounding Box .....	۵۱
شکل ۳-۲ روشهای تغییر اندازه.....	۵۲
شکل ۵-۱.....	۶۸
شکل ۵-۲.....	۶۹
شکل ۵-۳.....	۷۰
شکل ۵-۴.....	۷۲

شکل ۵-۵.....۷۳

شکل ۵-۶.....۷۳

شکل ۵-۷.....۷۴

## فصل ۱:

### مقدمه



## ۱-۱: آشنایی با OCR

پردازش و آنالیز تصاویر می‌تواند به عنوان یک ساختار کاربردی و تکنیکی جهت تسخیر کردن، تصحیح کردن، زیاد کردن و تغییر شکل دادن تصاویری که مشاهده می‌شود تعریف کرد.

در هنگام استفاده از تکنولوژی OCR، نرم افزار مربوطه تصویر را تجزیه و تحلیل می نماید و شکل حروف را بر اساس پراکندگی نقاط در تصویر، شناسایی می کند. متون می توانند حتی دارای عکس هم باشند که سیستم آنها را تشخیص داده، کنار می گذارد.

یک سیستم OCR به ما این امکان را می دهد که یک کتاب و یا یک مقاله را مستقیماً به یک فایل الکترونیکی تبدیل نماییم و آن را با کمک یک پردازشگر تغییر دهیم. استفاده از سیستم OCR یک راه کارآمد برای تبدیل متون به فایل‌های داده است که ممکن است تایپ کردن آنها زمان زیادی به طول بیانجامد. این تکنولوژی مدتهاست که به وسیله کتابخانه ها و سازمان های دولتی برای دستیابی الکترونیکی

سریع به مدارک حجیم به کار می رود و از لحاظ سرعت و هزینه روش مناسبی است. OCR از محدود زمینه های هوش مصنوعی است که می توان در عمل به آن تکیه کرد .

از متداول ترین کاربردهای این تکنولوژی می توان جداکردن

نامه ها در مراکز پست، دسته بندی چکها

بانک، تبدیل متن به صدا برای استفاده نابینایان، امکان دسترسی سریع به بایگانیهای عظیم و تبدیل کتابهای قدیمی کتابخانه های به فرمتی مناسب برای ذخیره در کامپیوتر را نام برد.

هدف از این عملیات زیاد کردن کیفیت نسبی اطلاعاتی است

که بعداً استخراج خواهد شد. تغییر شکل در حقیقت عملیاتی است که بر روی تصاویر ورودی انجام می گیرد تا خروجی به یک عدد که اطلاعاتی از تصویر است بدهد. تکنیکهای متفاوت تولید تصاویر در دسترس می باشد با وجود این تکنیکهای پردازش اطلاعات به عملیات

گسترده ای اطلاق می شود که می تواند بر روی اشکال و فرمتهای مختلف تصاویر عمل کند.

ابتدا تصاویر جهت از بین بردن اعوجاجی که در مرحله جمع آوری تصاویر ایجاد می شوند تشخیص می گردند و تصحیح سپس پردازش تصاویر یک عملیاتی است که یک سری کاربردها را ارجحیت می بخشد. تکنیکهای جدید یا پروسه های جدید که جهت حل مسائل مشخص وجود دارند از میان کاربردهای مختلف انتخاب می شوند.

پردازش تصاویر در سطوح مختلف انجمن های مدرن انجام می گیرد. روشها و تکنیکها در همه این سطوح به یک شکل می باشد و روشهایی که در مقوله مراقبت سلامتی انجام می گیرند از سایر روشهای پردازش تصاویر که در علوم مختلف و صنعت انجام می گیرد اقتباس می شوند. جهت توضیح دادن عمومیت پردازش تصاویر، تعدادی از کاربرهای پزشکی و غیر پزشکی در زیر آمده شده است.

## ۲-۱: پردازش تصاویر ماهواره:

تصاویر زمین و سایر سطوح جهانی توسط دوربین‌هایی که بر روی سفینه‌های فضایی و ماهواره‌ها نصب شده گرفته می‌شود. هنگامی که از سیارات دیگر تصویر گرفته می‌شود، هدف مطالعه سطوح آنها جهت بدست آوردن آگاهی درباره تاریخچه و مشخصات آن می‌باشد. شبیه‌سازی ماهواره‌ای زمین جهت کارهایی از قبیل کارهای فلاحتی- زراعتی، هواشناسی، محیط اطراف، پردازشهای نظامی در بین سایر موارد می باشد.

## ۳-۱ پردازش تصویر در صنعت و پزشکی:

تصاویر بخشهایی از سطوح زمین جهت کاهش تصویر و کوچکتر کردن، بازرسی و یا جهت نقشه‌کشی گرفته می‌شود.

- چاپ کردن و دست نوشته‌ها:

تشخیص کاراکترهای چاپ یا نوشته شده از کاربردهای جدید پردازش تصاویر است. در بین کاربردهای دیگر می‌توان به خواندن اتوماتیک کدهای پستی و کدهای بانکی اشاره کرد.

• بارکد:

مسئول واریسی در بیشتر سوپرمارکتها دارای وسایلی هستند که بارکد موجود بر روی کالاها را خوانده و تفسیر می‌کند. از استفاده‌های بارکد می‌توان به نمونه‌های مدارک، نمونه‌های بیوشیمیایی و نمونه بافت در بیمارستان اشاره کرد.

• رادیولوژی:

در مبحث پزشکی، رادیولوژی یکی از کاربردهای اصلی تکنیک پردازش تصاویر است. کاربردهای آن نه تنها در اشعه X می‌باشد بلکه کاربردهایی از قبیل مقطع‌بندی کامپیوتری PET, MRI, CT , ultra sound , تمام نتایج به صورت تصاویر هستند که باید بعداً تفسیر شوند.

در میان این کاربردها، بازسازی سه‌بعدی تصاویر یکی از موضوعهای مورد علاقه است.

- مجموعه خصوصیات کروموزومی موجودات:

از آنجا که تصاویر کروموزومی یکی از ساختارهای ساده می باشد، جفت‌شدگی کروموزومی و بازسازی خودکار آنها یکی از جدیدترین کاربردهای پردازش تصویر در پزشکی است.

- سلول شناسی و بافت شناسی:

تفسیر اتوماتیک و کوانتیزه تصاویر میکروسکوپی از کاربردهای دیگر پردازش تصاویر در پزشکی است.

## فصل ۲:

### مراحل OCR

## ۱-۲: سیستمهای بازشناسی حروف

در این فصل، پس از ذکر تاریخچه ای از OCR، مراحل تبدیل تصویر به متن را شرح داده و ورودی و خروجی هر مرحله را بیان می کنیم.

### • تاریخچه OCR

مبحث بازشناسی متون، همانند مبحث تشخیص صحبت از اوایل ظهور مبحث شناسایی الگو مطرح بود. در ابتدا شناسایی آنها آسان به نظر می رسید ولی در عمل بر خلاف انتظار بعد از یکسری پیشرفت کوچک، این مسأله بامشکلات بزرگی روبرو شد و توجه محققین به سمت سایر زمینه های شناخت الگو معطوف گشت. با پیدایش وسایل ارتباط جمعی الکترونیکی، نیاز شدیدی به سیستمهایی احساس شده است می شد که بتوانند با دقت و سرعت، اطلاعات نوشتاری موجود را خوانده و ذخیره نمایند.



در سال ۱۹۲۹ Taushech در آلمان و در سال ۱۹۳۳ Handle در آمریکا ابداعاتی در زمینه OCR ثبت نمودند. اینها اولین ایده های شناسایی حروف هستند. این روش، تطبیق الگو نام دارد و به این صورت کار می کند که به هر حرف، نور تابیده می شود و نور باز تابیده شده از حروف، از قالب های مکانیکی عبور داده می شود. هرگاه که نوری از قالب عبور نکرد، حرف تشخیص داده می شود.

در عمل تا زمان به عرصه آمدن کامپیوتر یعنی سال ۱۹۵۰ سیستم مطلوبی ساخته نشد. اولین کامپیوتر تجاری UNIVAC در سال ۱۹۵۱ در آمریکا شروع به کار کرد. در این زمان بود که ایده OCR به عنوان یک پدیده قابل پیاده سازی پذیرفته شد. از این زمان به بعد تحقیقات گسترده ای بر روی این تکنولوژی انجام گرفته است و در نتیجه سیستمهای تجاری عرضه شده اند که دارای قابلیت های خوبی هستند. سیستمهای تجاری عرضه شده را می تواند به سه نسل تقسیم بندی نمود.

سیستمهای نسل اول در اوایل دهه ۱۹۶۰ به بازار آمدند که نتیجه تحقیقات دهه ۱۹۵۰ بودند. آنها فقط قابلیت تشخیص کاراکترهای خاص با اندازه و فونت مشخصی را داشتند. در آنها از روشهایی استفاده شده بود که نسبت به تغییر جای کاراکترها، اندازه و دوران آنها فوق العاده حساس بودند. شاخص ترین سیستم این نسل، NCR 420 بود که می توانست اعداد و پنج نماد دیگر را تشخیص دهد.

سیستمهای نسل دوم در اواسط دهه ۱۹۶۰ به بازار آمدند و می توانستند بعضی از کاراکترهای دست نویس مانند اعداد را تشخیص دهند. RETINA یکی از این سیستمها بود.

سیستمهای نسل سوم مربوط به اواخر دهه ۱۹۶۰ هستند که باهدف تشخیص کاراکترهایی با کیفیت چاپ پایین و همچنین متون دست نویس لاتین ساخته شدند. این نسل تا به امروز هم ادامه دارد و فعالیتهای گسترده ای در این زمینه در حال انجام است. امروزه

سیستمهای OCR قادر به تشخیص دقیق کاراکترهای تایپی لاتین با انواع فونتها و در اندازه های متفاوت هستند (تا ۹۹٪) ولی هنوز در مورد تشخیص متون دست نویس لاتین و یا فونتهایی که در آنها از خطوط خمیده استفاده می شود (مثل فارسی و عربی) مشکلات زیادی وجود دارد.

روشهای بسیاری تا به امروز برای شناسایی متون فارسی و عربی به کار رفته اند. در سال ۱۳۵۹ به دنبال ابداع سیستمی برای شناسایی دست نوشته های عربی موسوم به IRAC توجهات به این زمینه نوین در قلمرو بازشناسی الگو معطوف شد. به دنبال آن سیستم دیگری برای تشخیص کلمات تایپ شده عربی پیشنهاد شد که در آن ابتدا کلمات به حروف تفکیک شده و سپس توصیف کننده های فوریه دوره خارجی حروف، مبنای شناسایی آنها قرار می گرفتند. به موازات آن دو روش ساختاری مطرح شدند:

- یکی مبتنی بر ایده دنبال کردن کانتور برای شناسایی حروف مجزای عربی و دیگری بر اساس استفاده از محلهای تقاطع و انشعاب پاره خطها برای شکستن کلمات به زیر حرفها و دسته بندی آنها به کمک ویژگیهای هندسی و توپولوژیکی.

در سال ۱۳۶۰ روشی برای شناسایی فونتهای بزرگ فارسی ابداع شد که از ویژگیهایی نظیر وجود حفره و دهنه برای شناسایی حروف بهره می جست. در سال ۱۳۷۲ دو سیستم جهت شناسایی حروف در متن فارسی تایی ارائه گردید که یکی با روش شکل شناسی و دیگری بر اساس توصیف کننده های فوریه عمل می نمود. امروزه با کمک شبکه های عصبی که قادرند ارتباط پیچیده ای بین ورودی و خروجی برقرار کنند، می توان حروف فارسی را شناسایی نمود. مرجع [۷] شامل پیوندهایی به چند پایان نامه انجام شده در زمینه OCR است. مرجع [۸] نیز برخی پایان نامه ها را بررسی و نتایج آنها را مقایسه نموده است.

پس از ذکر این تاریخچه، به شرح مراحل مختلف OCR می

پردازیم.

## ۲-۲. اسکن کردن صفحات

اسکن کردن، اولین مرحله کار یک سیستم OCR است. یک

پرتو نور به صفحه مورد نظر تابیده می شود. با حرکت این پرتو روی

صفحه و با تغییر مقدار سیاهی (و یا رنگ در اسکنرهای رنگی) شدت

نور بازتابیده شده تغییر می کند. نور بازتابیده شده به آرایه ای از سلول

های فتوالکترودیک برخورد کرده و به این ترتیب، یک فایل تصویری از

صفحه اسکن شده بدست می آید. از آنجایی که با دیدن این تصاویر می

توانیم محتویات صفحه را دریابیم، ممکن است این سوال پیش آید که

چرا باید زمان و هزینه تبدیل تصویر به متن را تقبل کنیم؟ در پاسخ باید

گفت:

- حجم فایل‌های تصویری بسیار بیشتر از فایل‌های متن است، حتی اگر

از روشهای فشرده سازی استفاده کنیم.

- تغییر در نحوه نمایش فایل‌های تصویری مشکل و یا عملاً "غیر ممکن است، مثلاً" تغییر فونت، در صورت امکان بسیار مشکل خواهد بود و یا ریز و درشت کردن تصویر می‌تواند به ناخوانا شدن حروف بیانجامد.

- تبدیل تصویر به اشکال دیگر ارائه، مثلاً "خواند متن برای نابینایان، بسیار مشکل است.

- امکان جستجوی عبارتی خاص و یا مقایسه بخش‌هایی از متن تصویری با متون دیگر عملاً "وجود ندارد.

مجموعه علت‌های بالا باعث شده تا تلاش‌های بسیاری برای تبدیل تصاویر به متن صورت گیرد. روش‌های دیگری، از جمله استفاده از دوربین دیجیتال، نیز برای ایجاد فایل‌های تصویری وجود دارد. این روش‌ها معمولاً در زمینه‌های دیگر تشخیص الگو (مثل تشخیص چهره) بکار می‌روند.

در مورد تبدیل متون کتابها و سایر نوشته ها، اسکن کردن مناسب ترین روش است. البته در بعضی کاربردها (مثل LPR) استفاده از این دوربینها ضروری است.

### **۳-۲. استخراج نواحی شامل متن و تفکیک متن به حروف**

یک فایل تصویری اسکن شده، فقط شامل متن نیست بلکه ممکن است شامل تصویر اشخاص و مناظر، نمودار و سایر تصاویر غیر متنی نیز باشد. سیستم OCR باید بتواند نواحی شامل متن را شناسایی و استخراج کند.

الف) اگر چیدمان صفحه اسکن شده مشخص باشد (مثل چک بانکی، پلاک خودرو، فرم ثبت نام داوطلبین آزمونها) این کار با روشی ساده و با قابلیت اطمینان بالا امکان پذیر است. مثلاً" در شکل (۱-۲) با استفاده از موقعیت مکانی مستطیل های سیاه کوچک حروف مورد نظر به سادگی استخراج می شوند.

ب) اما اگر چیدمان صفحه مشخص نباشد (مثل صفحات کتاب) باید از روشهای پیچیده تری بهره گرفت که بتوانند بر اساس تفاوتهای تصاویر متنی با تصاویر اشخاص و مناظر، متن را استخراج کنند. بدیهی است که در این کار امکان خطا وجود دارد.

یکی از روشهای مورد استفاده در این حالت، استفاده از هیستوگرام ( histogram) است. هیستوگرام تصاویر متنی معمولاً "حول دو رنگ (نوشته و زمینه) مقدار بالا دارند ولی تصاویر غیر متنی در بسیاری از رنگها، مولفه دارند. برای کسب اطلاعات بیشتر در مورد روشهای استخراج نواحی متنی می توانید به [۱] مراجعه کنید.



شکل ۱-۲ نمونه فرم اسکن شده

تصاویر متن پس از استخراج، بر اساس یک آستانه که توسط روشهای آماری بدست می آید ( [۹], [۱۱] ) به تصاویر سیاه و سفید (باینری) تبدیل شده و توسط فیلترهای مثل LP و Median نویزگیری می شوند [۶].

اگر سیستم OCR بخواهد خروجی مشابه با ورودی بدهد (مثلاً" در تبدیل کتابها که خروجی از لحاظ بصری شبیه صفحه اسکن شده است) باید چیدمان صفحه اسکن شده را ذخیره کند. در

سیستمهای دیگر (مثل سیستمهایی که خروجی آنها صوتی است یا سیستمهای LPR) این کار لزومی ندارد.

پس از انجام این مرحله، متون استخراج شده باید به تعداد محدودی واحد "قابل تشخیص اتوماتیک" تفکیک شوند. معمولاً این واحدها حروف تشکیل دهنده متن هستند، هرچند در برخی الگوریتمها از زیرواژه ها (مثل "با" و "لا" در کلمه "بالا") به عنوان واحد تشخیص استفاده می شود.

در برخی از کاربردها مثل فرم آزمونها (شکل ۲-۱) دو مرحله استخراج نواحی و تفکیک به حروف، مستقیماً انجام می شود ولی در سایر کاربردها باید از الگوریتمهای ویژه ای استفاده کرد. این کار در مورد حروف چاپی لاتین نسبتاً ساده است زیرا حروف به صورت مجزا در کنارهم قرار می گیرند و بین آنها فاصله خالی وجود دارد. در مورد متون فارسی و سایر زبانهایی که به صورت "سرهیم" نوشته می

شوند باید کارهای پیچیده تری صورت گیرد. (به عنوان یک مرجع خوب در این مورد به [۱] مراجعه کنید).

در صورت انجام موفق این مرحله، مجموعه ای از تصاویر بدست می آید که هر تصویر شامل یک حرف است. این تصاویر بعنوان ورودی به نرم افزار شناسایی کننده حروف داده می شوند. بسیاری از الگوریتمها تصویر رنگی یا خاکستری را نیز برای استفاده در پردازشها ذخیره می کنند.

#### **۲-۴. شناسایی حروف**

هدف از این مرحله عبارت است از تبدیل تصاویر "تک حرف" مرحله قبلی به کدهایی که نشانگر حروف مورد نظر است (مثل کد اسکی حرف متناظر).

به طور کلی روشهای شناسایی حروف بر دو نوع هستند:

- روشهای مبتنی بر تئوری تصمیم گیری

### • روشهای ساختاری

در روشهای نوع یک، ویژگیهای خاصی از تصاویر ورودی به صورت کمی اندازه گیری و استخراج شده و به صورت درایه های یک بردار نرم افزار شناسایی کننده را تشکیل می دهند. به این کار Feature Extractor می گویند. این ویژگیها می توانند شدت نقاط تصاویر خاکستری (یا مقدار نقاط تصویر باینری) ،  $projection$  افقی یا عمودی تصویر، میانگین، واریانس و یا نتایج اجرای یک فرایند پیچیده روی تصویر باشند. نرم افزار با توجه به بردار ویژگی، به هر یک از کلاسه های تعیین شد، امتیازی نسبت می دهد. کلاسهایی که بیشترین امتیاز را کسب کند، خروجی نرم افزار است. مثلاً" در تشخیص حروف لاتین، هر کدام از حروف A تا Z و a تا z یک کلاس هستند و نرم افزار هر بردار ورودی را به یکی از این کلاسه ها نسبت می دهد. یکی از مهمترین روشهای این نوع، شبکه های عصبی است که در فصل بعد، عملکرد آن را بیان خواهیم کرد.

در روشهای نوع دو، حروف با کمک ویژگیهای ساختاری (نقطه، سرکش، دسته، حلقه،...) و نحوه قرار گیری این ویژگیها در کنارهم شناسایی می شوند.

برای آشنایی بیشتر با انواع روشهای نوع یک و دو می توانید به [۱] مراجعه نمایید. همچنین [۸] شامل بخشی در مورد بکارگیری بعضی از این روشها در کاربردهای عملی و مقایسه نتایج آنهاست. آنچه در این پایان نامه تاکید میشود روش شبکه های عصبی بدون استفاده از الگوریتمهای پیچیده استخراج ویژگی است.

نشان خواهیم داد که شبکه های عصبی که حروف جایی را بدون بهره گیری از ویژگیهای پیچیده، تا حد قابل قبولی شناسایی می کنند، نمی توانند این کار را در مورد حروف دست نویس انجام دهند و باید ویژگیهای پیچیده ای استخراج شوند.

## ۵-۲: ارائه نتایج

هدف از این مرحله که مرحله پایانی کار سیستم OCR است، ارائه کدهای تولید شده در مرحله قبل به فرمت مناسب است. مثلاً "اگر هدف از OCR، تبدیل کتاب و مجله باشد، سیستم باید با استفاده از کدهای اسکی بدست آمده و اطلاعات چیدمانی که در مراحل قبل ذخیره شده، صفحه اسکن شده را شبیه سازی کند طوری که از نظر بیننده، تفاوتی بین صفحه کاغذ و فایل تصویر وجود نداشته باشد ولی از نظر رایانه، به جای تصویر حروف، کد آنها جایگزین شده باشد. به همین ترتیب اگر هدف، تبدیل تصویر به صوت است باید کدهای بدست آمده توسط الگوریتمهایی مثل Text to Speech API به اصوات شنیداری تبدیل شوند.

### فصل ۳:

### شبکه های عصبی

## ۶-۲: شبکه های عصبی

در صفحات پیشین شبکه عصبی را به عنوان یک روش متداول در شناسایی حروف معرفی کردیم. در این فصل ساختار و عملکرد شبکه های عصبی را بررسی کرده و در مورد ساختار و الگوریتم یادگیری یک نوع خاص شبکه عصبی (شبکه Multilayer Perceptron) که در این پایان نامه به کار رفته است، توضیح خواهیم داد.

### ۳-۱. آشنایی با شبکه عصبی

شبکه عصبی سیستمی است متشکل از تعدادی واحد ساده به نام نرون که با اتصالات وزن دار به هم متصل شده اند. با تغییر دادن وزن این اتصالات در فرایندی به نام یادگیری، شبکه می تواند توابع پیچیده ای را پیاده سازی کند. به طور کلی برای استفاده از شبکه عصبی، ابتدا ساختار مناسب را انتخاب می کنند (بخش ۳-۲ را ببینید)



سپس شبکه را آموزش می دهند تا وزن اتصالات (که در ابتدا دارای مقادیر پیش گزیده و یا تصادفی هستند) طوری تنظیم شوند که برای تعداد محدودی ورودی نتیجه مطلوب حاصل شود. سپس بقیه ورودیها را به شبکه آموزش دیده می دهند و نتایج حاصل را دریافت می کنند. از ویژگیهای شبکه های عصبی می توان به موارد زیر اشاره کرد [۷],[۱۱]:

- می توانند توابع پیچیده و غیر خطی را مدل سازی کنند.
- می توانند بر اساس نمونه ها یاد بگیرند و احتیاجی به دانستن تابع درونی شبکه بصورت تحلیلی نیست.
- به دلیل فعالیت موازی نرونها، شبکه های عصبی پس از یادگیری، می توانند با سرعت بالایی ورودیها را پردازش کرده و نتایج را بیرون دهند. تحمل پذیری بالایی نسبت به نویز ورودی دارند. در صورت خرابی یک یا چند نرون بازهم تا حد خوبی می توانند به کار خود ادامه دهند.

امروزه شبکه های عصبی در زمینه های مختلف پزشکی، دفاعی، صنعتی و... کاربرد دارند [۹]. در زمینه های گوناگون تشخیص الگو (مثل تشخیص صحبت، اثر انگشت، چهره و OCR) نیز شبکه های عصبی، بسیار به کار می روند.

ایده اولیه این شبکه ها، از شبکه های عصبی طبیعی گرفته شده است. در شبکه های طبیعی، همه فعالیت های ذهنی و حرکتی، با اتصال سلول های عصبی یا نرونها انجام می پذیرد. سرعت انتقال پیام الکتریکی در طول یک سلول و بین سلول های عصبی، نسبت به سرعت انتقال در سیستم های الکترونیک بسیار کمتر است (حداکثر تا چند متر بر ثانیه) و قابلیت پردازش آنها با کامپیوترها قابل مقایسه نیست ولی موجودات زنده فعالیت های بسیار پیچیده ای انجام می دهند که بسیاری از آنها قابل پیاده سازی با فن آوری فعلی نیست. این قابلیت را به تعداد زیاد این سلول ها و فعالیت موازی آنها نسبت می دهند.

شبکه های عصبی مصنوعی به منظور استفاده از این مزیتها ایجاد شده اند. هرچند به دلایل گوناگون، شبکه های عصبی مصنوعی از نظر ساختار و عملکرد، از شبکه های طبیعی فاصله گرفته اند [۷].

### ۱-۶-۲ انواع شبکه های عصبی

شبکه های عصبی را از چند نظر می توان تقسیم بندی کرد. از نظر ساختمان ، شبکه ها به دو نوع تقسیم می شوند.

(۱) Feedforward

(۲) Recurrent یا Feed Back

در نوع اول، مسیر انتقال اطلاعات همواره رو به جلو است ولی در نوع دوم، حلقه وجود دارد. شبکه های نوع دوم ممکن است ناپایدار باشند و یادگیری در این شبکه ها، بسیار کندتر است. این شبکه ها در تحقیقات، توجه زیادی به خود جلب کرده اند. ولی در کاربردهای عملی، شبکه های Feedforward سودمندتر بوده اند [۱۱].

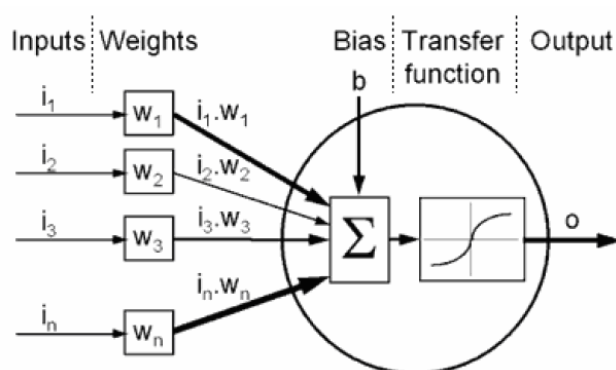
از شبکه های نوع اول می توان به Multilayer Perceptron و Hebbian و از شبکه های نوع دوم به Boltzman و Hopfield اشاره کرد. از نظر نحوه یادگیری شبکه ها به دو نوع Supervised و Unsupervised تقسیم می شوند.

در شبکه های نوع اول، مجموعه ای از ورودیها و خروجیهای متناظر آنها (خروجیهایی که انتظار داریم شبکه به ازای ورودیها داده شده، آن خروجی را تولید کند) به شبکه داده می شود. آنگاه توسط الگوریتمهایی مثل گرادیان کاهشی، وزن اتصالات طوری تنظیم می شود که تابع هزینه (معمولا "مجموع مجذورات تفاوت خروجی تولید شده توسط شبکه و خروجی مورد انتظار یا SSE) مینیموم شود.

در روشهای نوع دوم، اطلاعاتی از خارج به شبکه داده نمی شود و شبکه با استفاده از ویژگیهای مختلف ورودیها (مثل ویژگیهای آماری) خروجی می دهد. این روشها بر خلاف روشهای نوع اول معمولا "در کاربردهای تشخیص و طبقه بندی به کار نمی روند بلکه

بیشتر برای فشردن سازی و حذف redundancy و یا یافتن الگوهای خاص بین داده های ورودی استفاده می شوند . بحث در مورد انواع شبکه های عصبی بسیار گسترده است و ما به همین مقدار اکتفا می کنیم. برای کسب اطلاعات بیشتر در این مورد می توانید به [۱۰] مراجعه کنید.

## ۲-۲: نرون



شکل ۲-۲ یک نرون با تابع فعال ساز

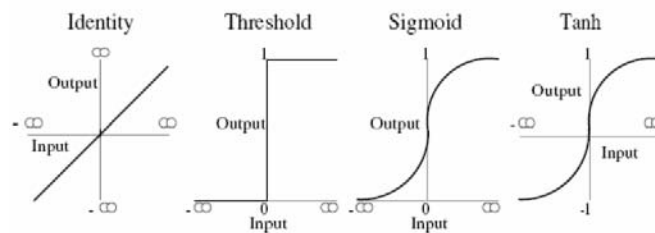
نرون واحد تشکیل دهنده شبکه عصبی است. قسمت‌های تشکیل دهنده یک نرون عبارتند از:

- مجموعه ای از ورودیها (i) که می توانند از خارج از شبکه تأمین شوند و یا خروجی سایر نرونها باشند. هر ورودی وزن متناظر دارد که قابل تغییر است (W)

- یک مقدار بایاس یا آستانه که همواره مقدار ثابتی است (معمولاً +۱ و گاهی -۱) ولی وزن آن قابل تغییر است. بایاس را می توان یک ورودی همواره ثابت فرض کرد.

- تابع ترکیب کننده که مجموع وزن دار ورودیها و بایاس  $(\sum i w_i + b)$  را محاسبه می کند.

- تابع فعال سازی یا انتقال که بر اساس مقدار خروجی تابع ترکیب، خروجی می دهد. این خروجی همان خروجی نرون است. در شکل ۲-۳ چند نمونه از توابع فعال سازی را مشاهده می کنید. برای مشاهده چند تابع دیگر می توانید به [۹] مراجعه نمایید.



شکل ۲-۳. چند تاج فعال سازی رایج

## ۲-۸. شبکه های Multilayer Perceptron و الگوریتم

### Backpropagation

Multilayer Perceptron (که از این به بعد به اختصار آن

را MLP می نامیم) پرکاربردترین نوع شبکه عصبی است [۱۱]. در این

نوع شبکه، نرونها به صورت لایه های متوالی در یک شبکه

Feedforward قرار دارند بطوریکه هر نرونی به همه نرونهای لایه بعد

متصل است ولی بین نرونهای یک لایه، ارتباطی وجود ندارد.

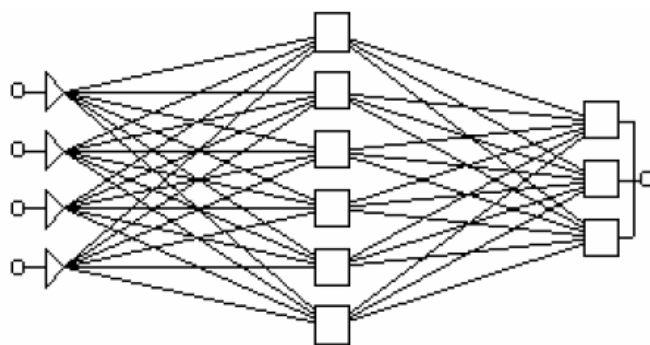
(شکل ۲-۴). لایه اول، لایه ورودی است. تعداد نرونهای این لایه، برابر

با تعداد ورودیها (به عبارتی دیگر، تعداد درایه های بردار ورودی) است.

لایه آخر لایه خروجی نام دارد و تعداد نرونهای آن با توجه به خروجی مورد انتظار تعیین می شود. مثلاً" در پیاده سازی توابع، یک نرون کافی است ولی در مسائل طبقه بندی، تعداد آنها برابر با تعداد کلاسهای خروجی است. لایه های بین این دو لایه، لایه های مخفی نام دارند.

می توان ثابت کرد که تمام مسائل قابل حل با شبکه های MLP، بایک لایه مخفی قابل پیاده سازی هستند هرچند وجود دو لایه مخفی می تواند از جهاتی سودمند باشد [۷]. تعداد نرونها در لایه های مخفی برخلاف دو لایه دیگر به سادگی قابل تعیین نیست بلکه باید با روش سعی و خطا و یا استفاده از تجربیات دیگران تعیین گردد [۴].





شکل ۴-۲ Multilayer Perceptron

فرض کنید شبکه دارای  $n$  اتصال وزندار باشد (شامل بایاسها). تابع خطای SSE را می توان به صورت سطحی در فضای  $n + 1$  بعدی در نظر گرفت. در هر لحظه با توجه به مقادیر وزنها، شبکه را می توان با نقطه ای در این فضا نشان داد. هدف از آموزش شبکه، هدایت این نقطه به سمت نقطه مینیموم سطح است. مشخص است که اگر در جهت عکس بردار گرادیان حرکت کنیم به سمت مینیموم خواهیم رفت (روش گرادیان کاهشی). البته ممکن است این نقطه مینیموم محلی و نه مینیموم کلی باشد. این مشکل، مشکل اصلی روش گرادیان کاهشی

است که ما در ادامه راه هایی را برای کم کردن احتمال بروز این مشکل ارائه خواهیم داد. اصولاً "یافتن ساختار بهینه و پارامترهای یادگیری شبکه با این مشکل مواجه هستند.

شبکه های MLP را غالباً "با الگوریتم Backpropagation

که مبتنی بر روش گرادیان کاهشی است آموزش می دهند. در این روش به طور iterative با اعمال ورودیها به شبکه و بدست آوردن تابع خطا، وزنهای را اصلاح کرده و مجدداً ورودیها را اعمال می کنیم و این کار را تا محقق شدن یکی از شرایط توقف ادامه می دهیم. مراحل این الگوریتم عبارتند از [۴]:

(۱) وزنهای را برابر با مقادیر تصادفی قرار می دهیم. این کار علاوه بر کمک به همگرایی، باعث می شود تا الگو ریتیم در هر بار اجرا شدن، از وضعیت جدیدی شروع شود و لذا در صورت ناموفق بودن اجرای آن (بخصوص در صورت افتادن در مینیموم محلی) امکان موفقیت در اجرای بعدی وجود داشته باشد.

۲) اولین بردار ورودی را به شبکه اعمال می کنیم (آغاز یک گذرالگوریتم).

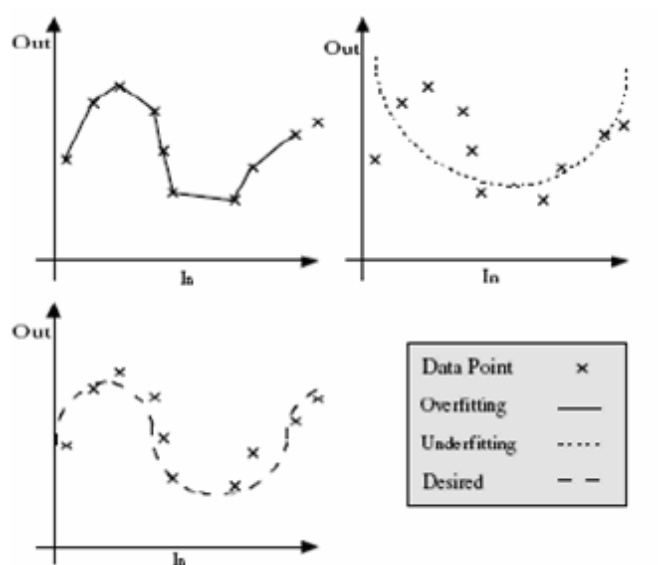
۳) بر اساس ورودیها، خروجیهای اولین لایه مخفی را محاسبه می کنیم. خروجیهای لایه دوم را بر اساس خروجیهای لایه اول محاسبه کرده و این کار را ادامه می دهیم تا خروجیهای شبکه (0 ها) بدست آیند.

#### ۲-۹. Overlearning

در بخش قبلی با الگوریتمی برای کاهش خطای شبکه آشنا شدیم. نکته قابل توجه این است که این الگوریتم خطا را روی مجموعه داده های یادگیری مینیموم می کند نه بر روی داده های آزمایش که مجموعه ای متفاوت هستند (معمولا" تعدادشان ناشناخته است و ما دقیقا" نمی دانیم چه ورودیهایی به شبکه اعمال خواهند شد). در نتیجه ممکن است شبکه به جای اینکه با استفاده از داده های یادگیری، تابع کلی را که ورودی و خروجی مورد انتظار را به هم مربوط

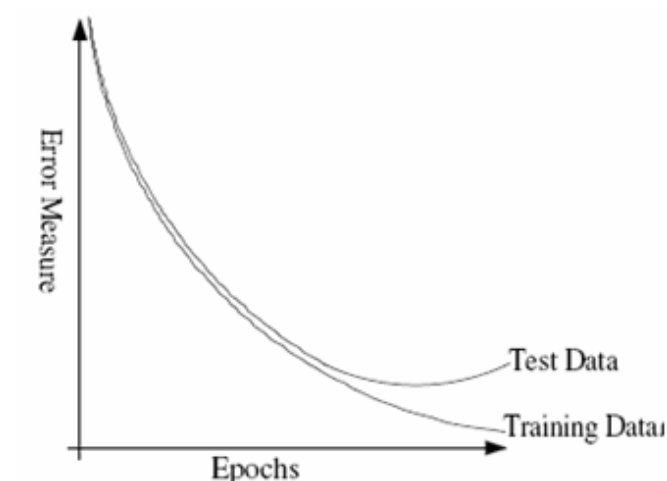
می کند تخمین بزند، تنها تابع مرحله یادگیری را پیاده سازی کند. هر چند این کار سبب کاهش خطای یادگیری می شود اما خطای آزمایش را که برای ما مهم تر است بیشتر می کند. به این پدیده Overlearning می گویند.

برای درک بهتر این پدیده می توان پدیده مشابهی (Overfitting) را در مورد تخمین چندجمله ای بر اساس چند نقطه داده شده بررسی کرد (شکل ۵-۲). در صورت overfit شدن، هر چند چندجمله ای پیچیده تر، با خطای کمتری نقاط را در بر گرفته ولی این چندجمله ای با چندجمله ای مطلوب، متفاوت است



شکل ۵-۲ overfitting چند جمله ای

هرچه تعداد نرونهای شبکه و تعداد گذرهای یادگیری بیشتر شود، خطای آموزش کمتر می شود ولی امکان overlearning و همچنین زمان یادگیری افزایش می یابد (شکل ۶-۲).



شکل ۶-۲ overlearning در شبکه عصبی

برای حل این مشکل، مجموعه ای از داده ها به نام داده های اعتبار را در نظر می گیریم.

این داده ها متفاوت از داده های آموزش بوده و در فرایند تنظیم وزنه های شبکه مورد استفاده قرار نمی گیرند. در هر گذر الگوریتم یادگیری داده های اعتبار را به شبکه اعمال کرده و خطای اعتبار را بدست می آوریم. اگر این خطا هم مثل خطای یادگیری کاهش پیدا

کرده بود، الگوریتم را ادامه می دهیم. در غیر اینصورت overlearning رخ داده و یادگیری را متوقف می کنیم. این حالت، نشان دهنده قدرت بیش از حد شبکه است. پس تعداد نرونهاى لایه مخفی را کاهش می دهیم و یادگیری را از نو شروع می کنیم. اگر الگوریتم بدون رخ دادن overlearning و بدون رسیدن به خطای یادگیری مطلوب به پایان رسید، تعداد نرونهاى لایه مخفی را افزایش می دهیم [۱۱].

در این فصل با شبکه های عصبی به عنوان یک راه برای شناسایی حروف آشنا شدیم. شبکه های عصبی MLF را که در کاربردهای تشخیص الگو رایج هستند معرفی کردیم و در مورد نحوه یادگیری آنها توضیح دادیم. با اتمام این فصل، بخش مقدمات نیز به پایان می رسد. در بخش بعد به کارهای انجام شده در این پایان نامه می پردازیم

## فصل ۴:

### شرح یک پروژه انجام شده



#### ۴-۱: شرح پروژه

این فصل شرح مختصری به یک نمونه کار انجام شده توسط آقای امید یگانه (مرجع شماره [1]) را دارد.

عملیات انجام شده در این پروژه در دو مرحله صورت می گیرد. مرحله اول شامل پردازش تصاویر و تهیه ورودی (فایل ویژگی) برای شبکه عصبی و مرحله دوم شامل ساختن و آموزش شبکه است. در این فصل مرحله اول یعنی تهیه ورودی شبکه عصبی را بررسی می کنیم. در ابتدا در مورد مشخصات فایل های تصویری مورد استفاده و سپس در مورد نحوه تبدیل تصاویر به بردار ویژگی و قراردادن بردارها در فایل ویژگی (به عنوان ورودی مرحله دوم) توضیح می دهیم.

#### ۴-۲. تصاویر مورد استفاده

فایل های تصویر استفاده شده در این پروژه ، برگرفته از بانک تصاویر حذف ۸۴ هستند. این بانک تصاویر، توسط شرکت اندیشه نرم افزار پایا و به سفارش شورای اطلاع رسانی کشور تهیه شده است. این

حروف دست نویس فارسی بانک شامل دویست هزار تصویر از حروف مجزای دست نویس فارسی است که از فرم ثبت نام داوطلبان ورود به مراکز استعدادهای درخشان (سمپاد) در مقاطع راهنمایی و دبیرستان استخراج شده اند.

مشخصات فایل‌های تصویری استفاده شده به شرح زیر است :

- Grayscale bitmap ( bmp ) .

- dpi ۳۰۰ .

- 77×95 pixel

### **۳-۴. تبدیل تصاویر به بردارهای ویژگی**

ابتدا تصاویر خاکستری، سیاه و سفید می شود. سپس

Bounding Box حروف موجود در تصویر (کوچکترین مستطیل در

بردارنده حرف) پیدا شده و تصویر بریده می شود (شکل ۱-۳).



شکل ۱-۳. یافتن Bounding Box و بریدن تصویر

الف: تصویر اولیه ب: تصویر سیاه و سفید پس از بریده شدن

ابعاد تصویر بریده شده را با دو روش به اندازه  $15 \times 15$  تبدیل می کنیم:

- روش اول، تغییر ابعاد بدون حفظ نسبت آنها است. این روش، همان روش استاندارد تغییر ابعاد است. با این روش حرف موجود در تصویر بریده شده، تمامی فضای تصویر جدید را پر می کند ولی چون ابعاد تصویر بریده شده یکسان نیستند، یک بعد بیش از بعد دیگر تغییر می یابد و این سبب تغییر شکل حرف می شود (شکل ۲-۳).

- روش دوم، تغییر ابعاد با حفظ نسبت آنها است. با توجه به اینکه بعضی از حروف (مثل "ق" و "و" و "ت") در اثر تغییر ابعاد به روش اول ممکن است شبیه به هم شوند، یک بار نیز با حفظ نسبت ابعاد تصویر را

تغییر می دهیم [۳]. در این روش هر دو بعد تصویر را به یک نسبت تغییر می دهیم بطوریکه بعد بزرگتر برابر با ۱۵ و بعد دیگر کمتر از ۱۵ شود (مثلاً) اگر ابعاد تصویر بریده شده،  $h$  و  $w$  باشند که  $h < w$ ، تصویر جدید دارای ابعاد  $15 * 15$  است پس با اضافه کردن متقارن سطرها یا ستونهای سفید به بعد کوچکتر، آن را به ۱۵ نقطه می رسانیم بطوریکه حرف در وسط تصویر باشد (شکل ۲-۳).



شکل ۲-۳. روشهای تغییر اندازه

الف: تصویر اولیه ب: بدون حفظ نسبت ج: با حفظ نسبت

در انتها تصویر تولید شده در مرحله اول را روی تصویر مرحله دوم می گذاریم تا تصویری به ابعاد  $30 * 15$  تشکیل شود و این تصودر را به

برداری با ۴۵۰ عضو که مقدار هر عضو صفر یا یک است تبدیل می کنیم. همانطور که قبلاً هم اشاره کردیم، ما از روشهای پیچیده تر استخراج ویژگی استفاده نمی کنیم و به تغییر اندازه اکتفا می نماییم. برای آموزش شبکه عصبی، باید به طریقی به شبکه عصبی اطلاع دهیم که هر بردار ورودی نشان دهنده چه حرفی است. هرچند در مرحله آزمایش در یک سیستم واقعی، از قبل نمی دانیم بردار ورودی نشان دهنده چه حرفی است (اگر می دانستیم احتیاجی به OCR نداشتیم) ولی در این پروژه باید این اطلاعات را داشته باشیم تا بتوانیم درصد تشخیص صحیح شبکه را در صورت همگرا شدن بدست آوریم. در هر صورت، باید جواب صحیح مورد انتظار را هم به شبکه بد هیم. پس عضو ۴۵۱-ام را به بردار ورودی اضافه کنیم. مقدار این عضو، برابر کد متناظر با حرف مورد انتظار است. حال همه بردارهای تشکیل شده را (هر بردار برای یک تصویر) در یک فایل متنی

(.txt) می نویسیم. به این ترتیب فایل ویژگی که ورودی مرحله بعد است، ایجاد می شود.

در این فصل مرحله اول عملیات پروژه را توضیح دادیم و نحوه ایجاد فایل ویژگی را بررسی کردیم. در حله بعد مرحله دوم را توضیح داده و شبکه ای عصبی برای تشخیص حروف ساخته و سعی می کنیم آن را آموزش دهیم.

#### **۴-۴: آموزش شبکه عصبی**

در این فصل درباره مرحله دوم تشخیص حروف، یعنی آموزش شبکه عصبی با ورودیهای تولید شده در مرحله اول، توضیح می دهیم. در بخش نخست، طراحی ابتدایی شبکه را بررسی کرده و خواهیم دید که این شبکه همگرا نخواهد شد. در بخش دوم، تغییرات و ساده سازیهای را که به منظور رفع مشکل عدم همگرایی شبکه انجام دادیم، بیان خواهیم کرد (هرچند که این کارها نیز نتوانستند مشکل را برطرف

سازند). با توجه به عدم همگرایی شبکه در مرحله آموزش، مرحله آزمایش انجام نشده است.

#### ۴-۵. طراحی ابتدایی

شبکه به صورت MPL سه لایه (ورودی، خروجی و یک لایه مخفی) پیاده سازی شده است. تعداد نرونهای در لایه ورودی ۴۵۰ (به تعداد عناصر بردار ویژگی)، در لایه مخفی ۵۰ و در لایه خروجی ۳۲ عدد (به تعداد حروف فارسی) است.

تابع فعال سازی همه نرونهای لایه مخفی و لایه خروجی به این صورت است. (این تابع در نرم افزار MATLAB با پارامتر logsig معرفی می شود). تابع فعال سازی نرونهای ورودی، تابع خطی همانی است زیرا می خواهیم خروجی آنها، حاصل جمع وزندار ورودیها باشد. در حالت ایده آل، می خواهیم با اعمال هر حرف ورودی، نرون خروجی متناظر با آن، مقدار یک و بقیه نرونها مقدار صفر داشته باشند.

مثلاً اگر حرف ورودی "ج" بود، خروجی نرون ششم لایه خروجی برابر با یک و بقیه خروجیها صفر باشند.

با اعمال این ورودیها به شبکه، خروجیهای لایه ورودی به لایه مخفی و خروجیهای این لایه به لایه خروجی می رود. خروجی های شبکه مقداری حقیقی بین صفر و یک دارند (مقدار تابع  $\text{logsig}$  بین صفر و یک است). با توجه به اینکه کدام خروجی بیشترین مقدار را دارد، حروف ورودی تشخیص داده می شود (مثلاً اگر خروجی نرون ششم لایه خروجی بیش از مقدار خروجی سایر نرونهای این لایه بود، حرف ورودی "ج" تشخیص داده می شود).

برای آموزش شبکه از الگوریتم Backpropagation با learning rate دینامیک و بهره گیری از ممان استفاده کرده ایم (این الگوریتم در نرم افزار MATLAB با پارامتر  $\text{traingdx}$  مشخص می شود). سایر پارامترهای الگوریتم یادگیری عبارتند از:

- ثابت ممان ( $\alpha$ ) برابر با ۰.۹۵.



- تعداد گذارهای الگوریتم (epochs) برابر با ۵۰۰۰

- تابع خطای SSE کمتر از ۰.۲٪

مقادیر عددی فوق بدون بررسی گسترده و صرفاً با توجه به

استفاده از این مقادیر در پروژه ای دیگر (مثلاً [۳]) انتخاب آزمایش کردیم.

باید توجه داشت که نمی توان مقادیر SSE و epochs را به

میزان زیادی تغییر داد زیرا افزایش تعداد گذرها سبب طولانی تر شدن

زمان یادگیری می شود و افزایش مقدار خطای مجاز، هرچند ممکن

است به همگرایی شبکه کمک کند ولی سبب افزایش درصد تشخیص

اشتباه خواهد گشت.

برای آموزش شبکه دو نکته را در نظر گرفته ایم:

- آموزش، تدریجی انجام می شود. یعنی ابتدا الگوریتم را با ۲۰ بردار

ورودی اجرا می کنیم. پس از اتمام یادگیری، ۲۰ بردار دیگر را به ۲۰

بردار قبلی اضافه می کنیم و مجدداً "الگوریتم را با این ۴۰ ورودی اجرا می نماییم و هر بار چند بردار به مجموعه قبلی می افزاییم تا بردارهای ورودی به تعداد قبل توجهی برسند. سپس کل بردارها را اعمال می کنیم. تفاوت آموزش تدریجی با روش عادی این است که در این نوع آموزش پس از پردازش تعداد کمتری از ورودیها، وزنهای تغییر می یابند (گذرها کوتاه تر هستند). این مسأله بخصوص در مراحل اولیه الگوریتم که وزنهای مقدار تصادفی دارند مفید است زیرا در زمان کوتاهی، وزنهای به مقدار نهایی و مطلوب خود نزدیک می شوند.

در این پروژه ، پس از اینکه تعداد ورودیها به ۱۴۰ می رسید، شکل نمی توانست همگرا شود و حتی پس از اعمال تغییرات و ساده سازیها نتوانستیم این عدد را بیش از ۲۴۰ کنیم. این مشکل نشانگر آن است که بردارهای ورودی مناسبی به شبکه اعمال نکرده ایم. یعنی بین اعضای دو کلاس یکسان (مثلاً "دو حرف یکسان که توسط دو فرد نوشته شده اند) تفاوت زیادی وجود دارد ولی گاهی بین اعضای دو

کلاس متفاوت شباهت وجود دارد ( مثلاً "دو حرف " ن " و " ق " که توسط یک نفر نوشته شده اند).

- بردارهای ورودی به ترتیب تصادفی به شبکه اعمال می شوند. یعنی اینطور نیست که ابتدا "الف" ها و سپس "ب" ها اعمال شوند بلکه یک جایگشت تصادفی از حروف اعمال می شود. مثلاً ابتدا یک "ج" و سپس سه "ب" و بعد یک "ل". این کار دو علت دارد:

علت اول: این است که وقتی تعداد کمی از ورودیها را اعمال می کنیم (آموزش تدریجی) نباید همه از یک حرف باشند. علت دیگر: این است که اگر ابتدا "الف" ها و سپس "ب" ها پردازش شوند، شبکه به سمت "الف" ها تمایل بیشتری دارد یعنی خطای "الف" ها کمتر از خطای "ب" ها می شود.

#### ۴-۶. تغییرات و ساده سازیها

برای رفع مشکل عدم همگرایی، تعداد نرونهاى لایه مخفی را در چند مرحله افزایش دادیم تا به ۱۵۰ نرون رسید. در این حالت نیز

شبکه همگرا نشد. البته با توجه به تعداد نرونهای ورودی (۴۵۰)، ۱۵۰، تعداد کمی است ولی از آنجا که با افزایش نرونهای لایه مخفی، زمان و حافظه لازم برای یادگیری به شدت افزایش می یابد، این تعداد را بیش از این افزایش ندادیم.

اقدام دیگر، کم کردن تعداد کلاسهای خروجی با حذف حروف مشابه بود. مثلاً "از بین حروف "ب"، "پ"، "ت" و "ث"، تنها حرف "ب" به عنوان ورودی اعمال شد و بردارهای وروی بقیه این حروف، حذف گردیدند. با اینکار تفاوت بین کلاسهای مختلف افزایش یافته و لذا امکان همگرایی شبکه بیشتر می گشت. در صورت موفقیت این کار، می توانستیم همانند کار انجام شده در [۷]، با حذف نقطه ها و بنابراین لازم است تعداد نرونهای ورودی (به عبارت دیگر، تعداد اعضای بردار ویژگی) کاهش یابد. استخراج شکل اصلی حروف (که معمولاً بزرگترین مولفه همبندی شکل حرف است) کار تشخیص توسط شبکه عصبی را آسان تر سازیم.

همانطور که قبلاً" هم گفتیم، این تغییرات تنها سبب افزایش تعداد بردارها در آموزش تدریجی از ۱۴۰ به ۲۴۰ شد و نتوانست شبکه را همگرا کند.

هم زمان با اعمال دو تغییر فوق، در مقدار ثابت ممان هم تغییراتی صورت دادیم ولی این کار تأثیر خاصی در عملکرد شبکه نداشت. همچنین با توجه به اینکه مقدار خطای SSE بدست آمده بسیار بیشتر از ۰.۲٪ بود، مشخص بود که افزایش خطای مجاز نمی تواند به همگرایی شبکه کمک کند.

در این فصل نحوه آموزش شبکه و تغییرات اعمال شده برای حل مشکل عدم همگرایی آن را بررسی کردیم. با توجه به ناموفق بودن این کارها به این نتیجه رسیدیم که مشکل اصلی، در روش استخراج ویژگی است و باید از الگوریتم پیچیده تری استفاده کنیم. در فصل آخر به بیان نتایج حاصل از انجام این پروژه پرداخته و پیشنهاداتی را برای رفع مشکل عدم همگرایی ارائه می نماییم.

فصل ۵ :

## ویژگی های متون فارسی

## ۱-۵. خصوصیات متون فارسی

متون فارسی دارای خصوصیات متفاوتی نسبت به متون لاتین

هستند.

(۱) جهت نوشتاری این حروف از سمت راست به چپ است، در حالی

که اعداد در این زبان از سمت چپ به راست نوشته می شوند.

(۲) بیشتر حروف از خطوط منحنی شکل تشکیل می شوند و یا دارای

حلقه هایی هستند که اغلب در جهت عقربه های ساعت بسته می

شوند.

(۳) کلمات این زبان به صورت سرهم نوشته می شوند. این خصوصیت

یکی از مشکل سازترین عوامل در شناسایی این حروف است، زیرا

جداسازی حروف سرهم خیلی مشکل است. این همان مشکلی است که

در شناخت متون دست نویس لاتین هم وجود دارد.

(۴) بیشتر حروف فارسی دارای اشکال متفاوت بسته به جایگاهشان در

کلمه می باشند.

۵) بسیاری از حروف فارسی شبیه به هم بوده و تفاوت آنها فقط در

تعداد و مکان نقطه هایشان است (مثلاً "ب، پ، ت، ث")

۶) در مورد حروف سرهم، همپوشانی عمودی بعضی از حروف فارسی

نیز مسأله ای است که شناخت این حروف را با مشکل روبرو می نمایند.

این خصوصیات باعث می شود که برای بازشناسی متون فارسی

از روشهایی متفاوت با روشهای متداولی که برای شناسایی متون لاتین

استفاده می شوند، استفاده نماییم.

یک سیستم OCR فارسی در حالت کلی باید بتواند صفحات

را اسکن کرده، تصاویر قسمتهای مربوط به متن را از تصویر صفحه

اسکن شده استخراج و تصویر متن را به تصاویر حروف تفکیک نماید.

آنگاه هر حرف را شناسایی (تشخیص حرف از روی تصویر آن) و سپس

حروف شناسایی شده را به نحو مناسبی کنارهم چیده و خروجی

مطلوب را ارائه دهد.



جدا سازی حروف پیوسته کار بسیار مشکلی است و گرچه پیشرفتهایی در این زمینه انجام شده، هنوز سیستمی که بتواند با دقت قابل اطمینان این عملیات را انجام دهد ارائه نشده است. لذا در کارهایی که استفاده از OCR الزامی است (مثلاً "جمع آوری مشخصات داوطلبین در آزمونها)، متون را به صورت دنباله ای از حروف مجزا می نویسیم.

ورودی این سیستم، مجموعه ای از تصاویر است که هر کدام یک حرف دست نویس در بردارد. این تصاویر پس از پردازش مختصری (تبدیل به تصاویر سیاه و سفید و تغییر اندازه) برای آموزش شبکه عصبی بکار می روند. چنین پردازشی برای تشخیص حروف تایپی کافی است ولی در مورد حروف دست نویس خواهیم دید که شبکه علیرغم تلاشها و ساده سازیهای انجام شده، همگرا نمی شود. نتیجه خواهیم گرفت که در مورد حروف دست نویس، باید پردازش پیچیده تری روی تصاویر انجام شود.

فصل ۶ :

**برسی چند محصول OCR**

## ۱-۶ چند نمونه OCR

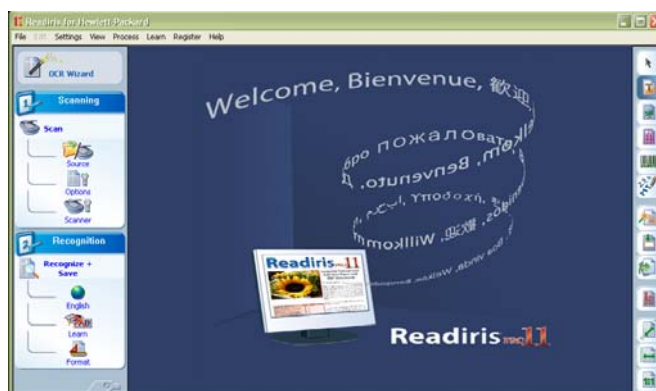
از نمونه نرم افزار هایی که در این زمینه وارد بازار شده اند می توان به Readiris Pro 11 و VajehShenas V.1 اشاره کرد که VajehShenas V.1 از نوع وطنی آن که به دست متخصصین ایرانی تولید شده است که در این فصل بر آن داریم این دو نرم افزار را آموزش دهیم

## ۲-۶ آموزش Readiris Pro 11



این نرم افزار قابلیت شناسایی اکثر زبان ها را دارا می باشد اما قادر به شناسایی زبان فارسی نیست. این نرم افزار قادر به شناسایی انواع عکس و PDF Word, Excel, و Email.....

هنگامی که روی آیکون آن دبل کلیک می کنیم شکل ۱-۵ را می بینیم

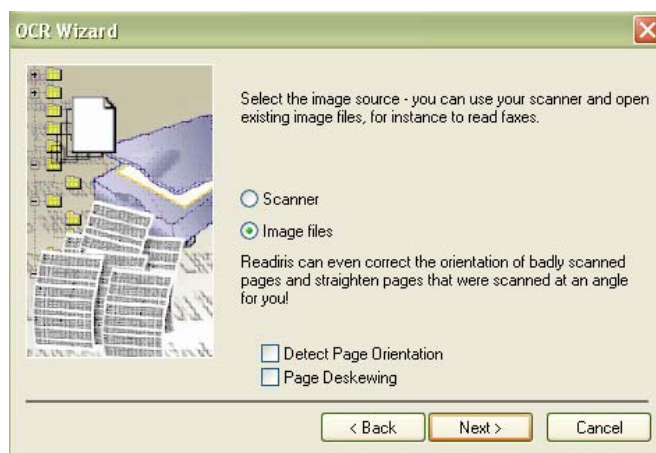


شکل ۵-۱

که به دو صورت wizard و دستی می توان با آن کار کرد:

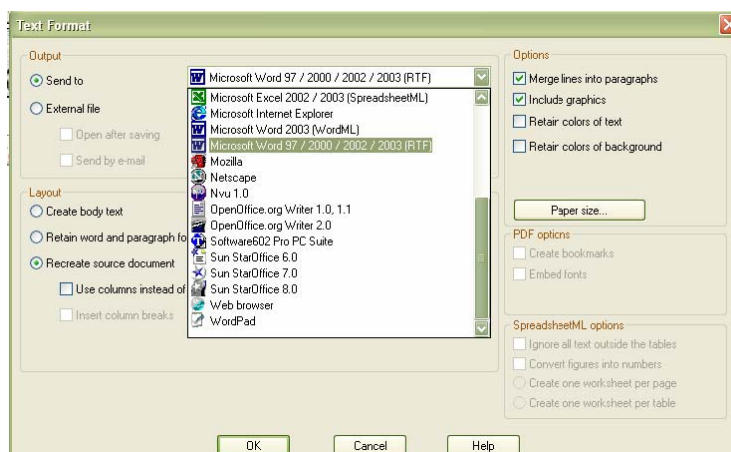
(۱) برای کار کردن به صورت ویزاردی روی متن ocr wizard در قسمت سمت چپ بالای کادر کلیک کرده پنجره‌های باز میشود که بر روی دکمه next کلیک می کنیم. در صفحه بعد (شکل ۵-۲) از ما منبع عکس را می خواهد که اگر عکس مورد نظر ما روی هارد قرار دارد

روی file image کلیک کرده در غیر اینصورت روی



شکل ۵-۲

گزینه scanner کلیک می کنیم سپس next را کلیک می کنیم.  
در صفحه بعد زبان نوشته را می پرسد. که در صورت دلخواه می  
توان آن را با دکمه change به راحتی عوض نمود. سپس با کلیک بر  
روی next به صفحه ای می رویم (شکل ۵-۳) که فرمت خروجی را از  
ما می پرسد که به دلخواه یکی را انتخاب مینیم ترجیحا "بهرتر است  
که microsoft word را انتخاب کنیم



شکل ۵-۳

سپس با کلیک کردن بر روی next به پایان عملیات می‌رسیم فقط در آخرین مرحله صفحه‌ای باز می‌شود که فایل عکس را انتخاب می‌کنیم که به صورت خودکار OCR فایل را مورد پردازش قرار می‌دهد سپس فقط نتیجه نهایی را در محل دلخواه کاربر ذخیره می‌کند.

(۲) اما به صورت دستی این گونه عمل می‌کنیم که از منوی فایل گزینه open را انتخاب کرده سپس عکس مورد نظر را انتخاب می‌کنیم. سپس (با توجه به عکس ۵-۱) بر روی گزینه recognize save

واقع در سمت چپ پایین صفحه را انتخاب می کنیم و اگر می خواهیم فرمت خروجی یا زبان را تغییر دهیم در پایین منوی recognize save به راحتی قابل تغییر است. سپس محلی که می خواهیم متن عکس ذخیره شود را انتخاب کرده و دیگر کار تمام است

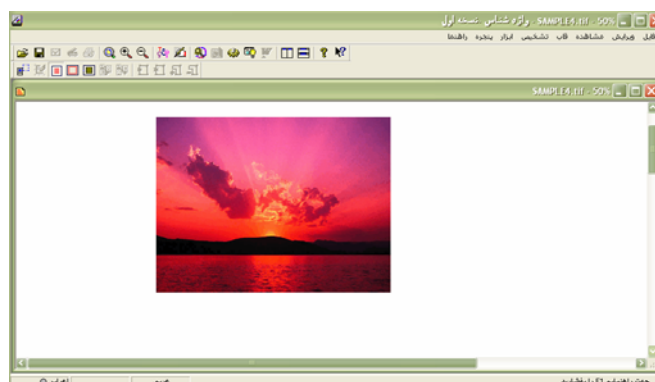
### ۳-۶ آموزش VajehShenas V.1



VajehShenas  
V.1

VajehShenas V.1 این نرم افزار کاملاً ایرانی توسط

شرکت هوش مصنوعی رایورز تهیه شده است که بر خلاف Readiris Pro 11 که فرمت های متنوعی را به عنوان ورودی می پذیرد VajehShenas V.1 فقط قادر به شناسایی بعضی از فرم های عکس مانند BMP TIF,PCX,ART را دارا می باشد.



شکل ۴-۵

(۱) برای WIZARD باید مراحل زیر را انجام دهیم ابتدا به منوی ابزار رفته و گزینه ی را هبر اتو ماتیک را انتخاب می کنیم که از پنجره باز شده نوع عکس را مشخص می کنیم (شکل ۵-۵) که از نوع ورد به کامپیو تر از طریق اسکن یا از نوع فایلی که درون هارد قرار دارد. سپس به دلخواه یکی را انتخاب و به صفحه بعدی می رویم و در این صفحه محل عکس را مشخص می کنیم. سپس گزینه دکمه بعدی را می زنیم در صفحه جدید که مهمترین صفحه می باشد از ما می پرسد که این عکس دارای چه زبانی می باشد مثلاً "فارسی یا



انگلیسی و.... در صورتی که متن عکس با گزینه پیش فرض مغایرت دارد حتما" باید آن را تغییر دهیم(شکل ۵-۶).



شکل ۵-۵



شکل ۵-۶

با زدن دکمه بعدی به صفحه جدیدی می رویم که مربوط به دوران عکس است که ما با آن کاری نداریم. سپس دکمه بعدی را زده که به آخرین مرحله که محل ذخیره سازی سند و فرمت آن را می پرسد کار دیگر تمام است (شکل ۵-۷)



شکل ۵-۷

۲) روش دستی این نرم افزار این گونه است. که از منوی فایل گزینه باز کردن را انتخاب کرده و عکس مورد نظر را انتخاب کرده و سپس از منوی قاب ، قاب بندی اتوماتیک را انتخاب کرده سپس به منوی تشخیص انتخاب موتور تشخیص رفته و زبان عکس که بسیار مهم است را با دقت انتخاب کرده و در آخر گزینه شروع از منوی تشخیص را سمهم این نرم افزار آن است که عکس با رزولو ۳۰۰dpi سیاه و سفید و با فرمت TIF گرفته شود و همین طور عمق رنگ آن ۱ باشد در غیر این صورت با مشکلات بسیاری مواجه هستید

#### ۴-۵ مقایسه VajehShenas با Readiris

در مقایسه ردیز و واژ شناس ذکر موارد زیر الزامی است:

- ردیز نرم افزاری بسیار تواناست که از فرمت های ورودی بیشتری نسبت به واژه شناس بهره می برد مانند pdf و..

- تعداد حروف شناسایی نشده توسط ردیز کمتر از واژه شناس در زبان انگلیسی است.
- واژه شناس بعد از شناسایی حروف دارای یک نرم افزار غلط یاب حرفه ای است که ردیز از آن بی بهره است
- ردیز دارای ظاهری شکیل تر نسبت به واژه شناس است

فصل ۷ :

## نتایج و پیشنهادات

در فصول گذشته دیدیم که شبکه عصبی طراحی شده، همگرا نشد. با توجه به عملکرد موفق شبکه های عصبی در این پایان نامه ها مشابه از جمله [۳] و [۸] و بخصوص [۷] که کاری مشابه با این پایان نامه را بررسی کرده، انتخاب روش شبکه عصبی برای تشخیص حروف و همین طور انتخاب نوع شبکه (MLP) انتخاب مناسبی است. با

توضیحاتی که در فصل های قبل ارائه شد، می توان نتیجه گرفت که روش مورد استفاده برای استخراج ویژگی از تصاویر ورودی، روش مناسبی نیست. برای این منظور باید روش دیگری در نظر گرفت که دارای سه ویژگی زیر باشد:

(۱) الگوریتم باید بتواند برای حروف مختلف، خروجیهایی با تفاوت قابل توجه بیرون دهد. این امر بخصوص در مورد حروف مشابه (مثل "ب" و "پ" مهم است).

(۲) الگوریتم باید بتواند برای نمونه های مختلف از یک حرف (مثلاً "دست خطهای مختلف از حرف "ج") خروجی های مشابه بدهد.

(۳) تعداد عناصر خروجی الگوریتم (که اعضای بردار ویژگی هستند) باید تا حد ممکن کم باشد

زیرا هرچه تعداد بیشتر باشد، تعداد نرونهاي لازم در لایه مخفی نیز بیشتر خواهد شد و افزایش تعداد نرونها و اتصالات، زمان و حافظه یادگیری و آزمایش نیز بیشتر خواهد شد. یک راه ساده و موثر برای

تحقق این شرط، حذف redundancy بین اعضای بردار ورودی با الگوریتمهایی مثل PCA است. یک ایده خوب برای کمک به تحقق دو شرط اول، برداشتن نقطه ها و سرکش حروف ورودی است.

ابتدا با پردازش تصاویر ورودی، بزرگترین مولفه همبندی تصویر را انتخاب کرده و با اعمال روشهای استخراج ویژگی و سپس بکارگیری شبکه عصبی، فرا-کلاس (مجموعه ای از چند حرف مشابه مثلاً "فرا-کلاس "ک-گ") متناظر با حرف ورودی را شناسایی می کنیم. سپس با توجه به شکل و مکان قرارگیری نقطه ها و سرکش، حرف ورودی را تعیین می کنیم. یک ایده خوب برای این مرحله، استفاده از یک شبکه عصبی دیگر است. به این صورت که یک شبکه عصبی اصلی برای شناسایی شکل اصلی حرف قرار می دهیم. درهر فرا-کلاس خروجی نیز یک شبکه عصبی دیگر قرار می دهیم. تصویری که نقطه ها و سرکش آن حذف شده اند به شبکه عصبی اصلی وارد شده و فرا کلاس آنها شناسایی می شود.

سپس تصویر اولیه و یا خروجی الگوریتمی که اطلاعات نقطه ها سرکش تصویر اصلی را دربر دارد، به شبکه عصبی مخصوص این فرا-کلاس وارد شده و حرف مورد نظر شناسایی می شود.

آنچه در تبدیل کلاسها به فرا-کلاسها مشکل ساز است، امکان متصل یا جدا بودن نقطه ها و سرکش در دست خطهای مختلف است که باید روشی برای شناسایی این نوع نقطه ها پیدا کرد.

## مراجع

[۱] امید یگانه، مونا. "جداسازی حروف تاییبی مستقل از نوع فونت و اندازه". پایان نامه کارشناسی ارشد، دانشکده مهندسی برق، دانشگاه صنعتی شریف، ۱۳۸۳.



[۲] اندیشه نرم افزار پایا، شرکت. "راهنمای بانک اطلاعاتی حذف ۸۴"

۱۳۸۴

[۳] بهنام قادر، احسان. "تشخیص کاراکترهای پلاکهای خودرو". پایان

نامه کارشناسی، دانشکده مهندسی

برق، دانشگاه صنعتی شریف، ۱۳۸۳.

[4] Beeman, Dave. "Multi-layer Perceptrons (Feed-forward Nets), Gradient Descent, and Back Propagation". University of Colorado, 2001.  
<http://ece-www.colorado.edu/~ecen4831/lectures/NNet3.html>

[5] Burrow, Peter. "Arabic Handwriting Recognition". Master of Science Thesis, School of Informatics, University of Edinburgh, 2004.

[6] Gonzalez, Rafael C., Woods, Richard E. "Digital Image Processing". Second Edition, New Jersey: Prentice-Hall, 2002.

[7] Johnston, Alexander. "Classifying Persian Characters with Artificial Neural Networks and Inverted Complex Zernike Moments". 2005.  
<http://www.doc.ic.ac.uk/teaching/projects/Distinguished05/AlexJohnston.pdf>

[8] Klassen, Tim. "Towards Neural Network Recognition of Handwritten Arabic Letters". A Project Submitted to the Faculty of Computer Science in Partial Fulfillment of the Requirements for the Degree of Master of Computer Science, Dalhousie University, 2001.

[9] MathWorks Inc. "MATLAB Help". MATLAB Version 7.0 (R14) Documentation, 2004.

[10] Sarle, Warren S. "Archive of Neural Network FAQ". 2002.  
<ftp://ftp.sas.com/pub/neural/FAQ.html>

[11] Stat Soft Inc. "Neural Networks". 2003.  
<http://www.statsoft.com/textbook/stneunet.html>

[12] Philippides, Andy. "Multi-Layer Perceptron (MLP)". Neural Networks Lectures, 2003.  
[http://www.informatics.sussex.ac.uk/users/andrewop/Courses/NN/NNs5\\_6\\_MLP.ppt](http://www.informatics.sussex.ac.uk/users/andrewop/Courses/NN/NNs5_6_MLP.ppt)

