Radial Basis Neural Networks (RBNs)

اسما سراج‌پور شوشتری، پردیس مهتدی، مجتبی اکبری

در شبکه‌های تک‌لایه‌ی پرسپترون (MLP) فقط linear separability داریم و برای ایجاد یک جداسازی غیرخطی، حداقل به یک لایه‌ی مخفی نیاز داریم. در این نوع خاص از شبکه‌ها، سیگنال ورودی به یک فرم دیگر تبدیل می‌شود به صورتی که بتوان آن را به شبکه داد تا جداساز خطی شود. (همان linear separable) اما RBNها از لحاظ ساختاری مانند شبکه‌های پرسپترون هستند.

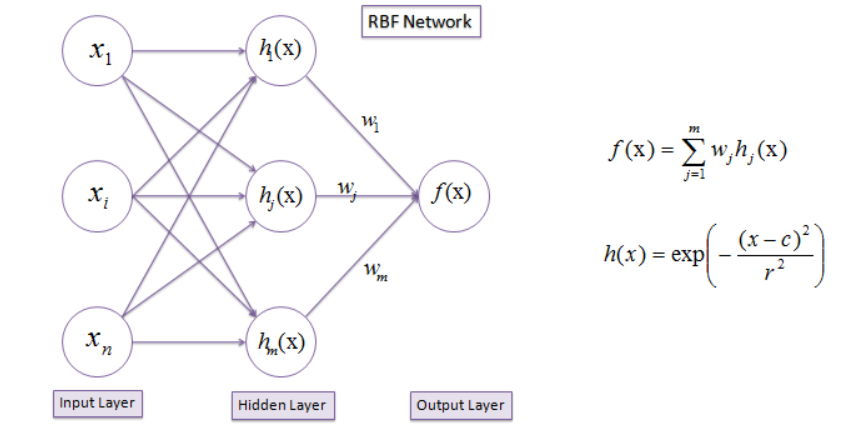
RBNNها از لایه ورودی، لایه خروجی و لایه مخفی بین آن‌ها تشکیل شدند و حتما باید فقط از ۱ لایه مخفی در آن‌ها استفاده شود که به آن لایه feature vector گفته می‌شود. در واقع RBN ابعاد این لایه که feature vector نامیده می‌شود را افزایش می‌دهد و با این کار linear separability آن افزایش می‌یابد.

Radial Basis چگونه عمل می‌کند:‌ یک receptor یا گیرنده تعریف می‌کنیم و نقشه‌های تقابل را اطراف آن رسم می‌کنیم. توابع گاسی اغلب در این شبکه‌ها استفاده می‌شوند.

Radial distance = ||x-t||

ϕ(r) = exp (- r²/2**σ²**)

نوشته‌ی بالا، تابع ردیال گاوسی را نشان می‌دهد.



برای یاددادن به این شبکه‌ها مراحل زیر طی می‌شوند:

۱- باید با back propagation لایه‌ی پنهان را یاد بدهیم.

۲- برای هر نود در لایه پنهان باید همان گیرنده‌ای که قبلا گفته شد را، پیدا کنیم. واریانس را هم باید پیدا کنیم.

۳- در فاز دوم یادگیری، وکتورهای وزن بین لایه مخفی و لایه خروجی را آپدیت می‌کنیم.

۴- در لایه پنهان هر نود بیانگر تغییری است که در radial basis function به وجود آمده است.

۵- در لایه‌های پنهان همه‌ی چیزهای عیرخطی‌مان را قرار می‌دهیم.

۶- با توجه به چیزهایی که گفته شد با کمک الگوریتم clustering توانستیم مرحله اول یادگیری را پیاده‌سازی کنیم. هرچندتا cluster center نیاز داشته باشیم تعریف می‌کنیم و با استفاده از این الگویتم این cluster centersها را محاسبه کرده که همین‌ها بعدا به عنوان <گیرنده> در نوروه‌های مخفی شناسایی می‌شوند.

۷- واریانس هر گیرنده = مجموع مربع فواصل بین گیرنده مربوطه و نزدیک‌ترین نمونه‌های cluster

مزایای RBNs:

۱-بسیار سریع‌تر از MLPها یاد می‌گیرند.

۲- می‌توان به راحتی مشخص کرد که هر گره چه عملکرد یا معنی‌ای دارد که در MLPها این کار دشوار است.

۳-classification در این نوع شبکه‌ها زمان بیشتری نسبت به شبکه‌های پرسپترون تک‌لایه می‌برد.

باید mlp classifier بزنیم. عکس‌ها را باید به شبکه‌ی عصبی بدیم. عکس‌ها لود می‌شن توی فایل mnist بعد فایل زیپ باز می‌شود و عکس‌ها را برمی‌دارد و شکلشان را عوض کرده و به هر کدام لیبل می‌زند (توی تابع mnist label) این‌ها داده‌ها هستند و با این داده‌ها شبکه یادگیری را شروع می‌کند.

عکس‌ها یک آرایه دوبعدی از اعداد هستند و یک سری از این عددها لیبل دارند و یک سری‌هاشون رو قراره پیشبینی کنیم. لیبل یک سریاشون مثلا می‌شه ۱ و ۰ و ...

حال باید شبکه عصبی را درست کنیم که منظور از mlp classifier همین است و چند لایه دارد. حالا این شبکه عصبی اون آرایه دوبعدی عکس رو می‌گیرد و forward propagation انجام می‌دهد و با این کار یک سری ضریب یا همان وزن به دست می‌آید. پس اگر یک بردار به یکی از نودها وارد شود، این ضریب را در بردار ضرب می‌کند و سپس آن را به لایه‌ی بعدی تحویل می‌دهد. البته قبل از این کار یک بایاس هم به آن اضافه می‌کند. بعد شبکه همینجوری جلو می‌رود تا خروجی بدهد (یک آرایه ده تایی خروجی می‌دهد مثلا می‌گوید عدد ۰ است یا ۲ است یا ۱۰ است یا چی) حالا در هر کدوم از این خونه‌ها یک درصدی قرار داده شده که میزان تعلق به اون عدد رو مشخص می‌کنه. اما در دفعه‌ی اول اکثر این اعداد اشتباه هستند.

پس back propagation انجام می‌دهد.

تابع اصلی ما (loss) تابعی است که نسبت به ورودی سنجیده می‌شود. در واقع نسبت به ورودی لایه‌ی یکی مانده به آخر است و آن هم نسبت به ورودی لایه دو تا مانده به آخر و همینجوری عقب می‌رود تا به ورودی اصلی برسد. حالا قرار است این مینیمم شود و چگونه این کار انجام می‌شود؟ با مشتق‌گیری. در این شبکه مشتق (مشتق هر چیزی می‌تواند باشد اما در این شبکه خاص اینگونه است)‌ تابع، اختلاف لیبل اصلی و اون عددای اشتباهی که به ما داده است (مثلا مشتقش می‌شود ۵)

مثلا در عکسی که پیوست شده است، loss مقدارش ۱.۶۷ است و هی کمتر می‌شود تا به صدم برسد.

پس یه بار رفت جلو، مشتق می‌گیریم می‌آریم عقب و همین یه ورودی جدید می‌شه و اینقد این کار رو تکرار می‌کنیم تا دقتش خوب شه.

Max iteration هم برابر است با ۱۰ یعنی نهایتا ۱۰ بار می‌تونیم بریم جلو و برگردیم.

حالا وزنای جدید چطوری به دست می‌آن؟ یه learning rate داریم که هربار در مشتق ضربش می‌کنیم و وزنای جدید به دست می‌آن. (۰.۱)

پس هی داریم وزنامونو آپدیت می‌کنیم.

Layer size یعنی چی؟ یعنی مثلا ۳۰ تا ورودی می‌گیریم ۳۰ تا هم خروجی می‌دیم.

\*\*از لایبرری mnist استفاده شده است.

\*\*در کد از numpy استفاده شده است.