بسم الله الرحمن الرحیم

تکلیف یک درس هوش محاسباتی

استاد عبادزاده

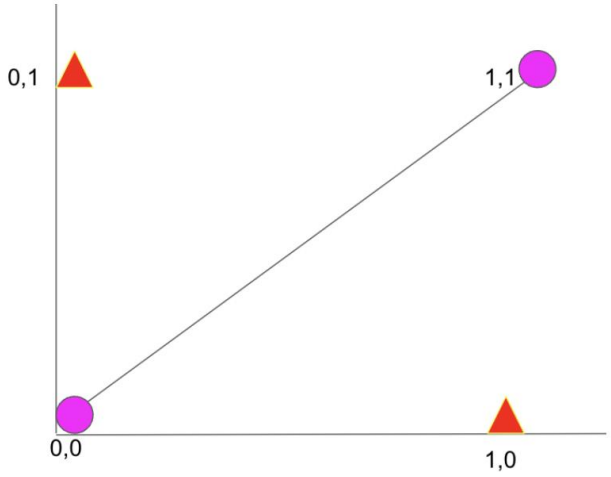
سید امیرمهدی میرشریفی

۹۸۳۱۱۰۵

۱) فرض کنید n متغیر boolean داریم که ورودی‌های ما هستند و می‌خواهیم حاصل XOR بین این متغیر ها را حساب کنیم. همچنین فرض کنید در تمامی قسمت‌های زیر توابع فعال‌سازی همه نورون‌ها تابع پله است.

a) آیا می‌توان تنها با یک پرسپترون این کار را انجام داد؟ توضیح دهید.

پاسخ: همانطور که در تصویر زیر مشاهده میکنید عملگر xor را نمیتوان با تنها یک پرسپترون پیاده سازی کرد زیرا با یک پرسپترون تنها میتوان قسمت بندی خطی را انجام داد.



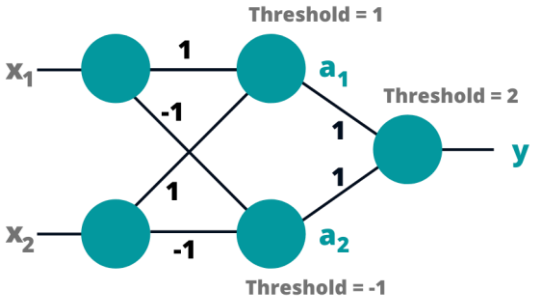
XOR

b) با کمک گرفتن از شبکه مورد نیاز برای تابع دو متغیره XOR ساختار شبکه ای را ترسیم کنید که تعداد نورون‌های آن از مرتبه $\theta(n)$ و تعداد لایه‌های پنهان آن از مرتبه $\theta(logn)$ باشد.

پاسخ: میتوان دو به دو ورودی ها را با یکدیگر xor و سپس باقی مانده ها را مجدد xor. این عملیات به ازای n ورودی نیاز به تتای lgn لایه دارد.

Y= ((x1 xor x2) xor (x3 xor x4)) xor .....

تصویر پایین یک شبکه عصبی است که عملیات xor بین دو ورودی را انجام می دهد که میتوان با استفاده از شبکه زیر که خروجی آن همراه یک خروجی از یک شبکه دیگر مجدد به عنوان دو ورودی دیگر به شبکه بعدی میرود میتوان شبکه مورد نظر را پیاده سازی کرد.



c) با کمک گرفتن از شبکه مورد نیاز برای تابع دو متغیره XOR ساختار شبکه ای را ترسیم کنید که تعداد نورون‌ها و تعداد لایه‌های پنهان آن از مرتبه $\theta(n)$ باشد.

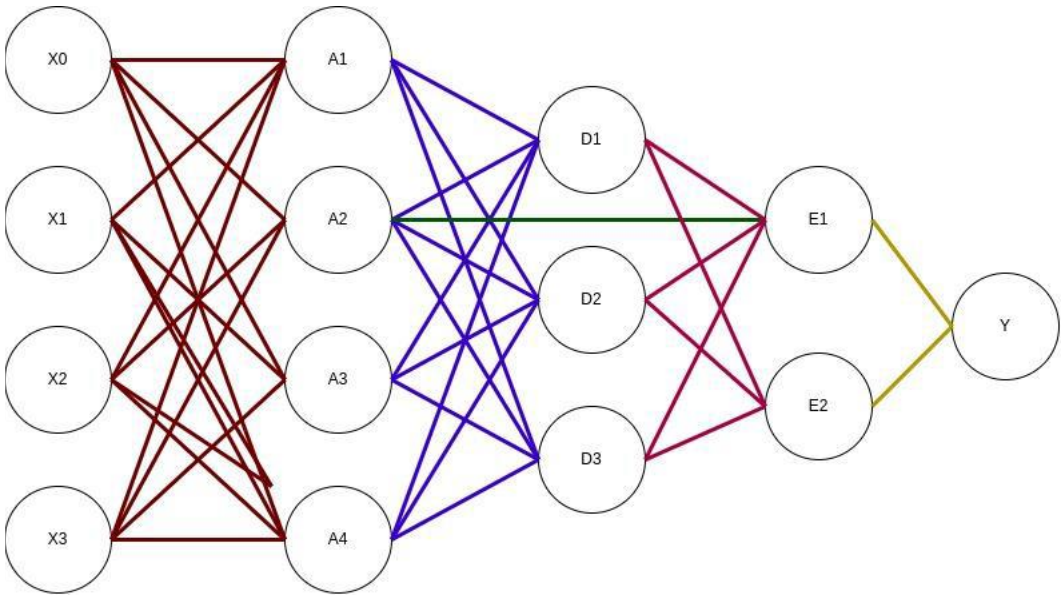
پاسخ: در این حالت بجای آن که دو به دو عملیات xor را انجام دهیم در ابتدا xor دو عنصر ابتدایی را انجام میدهیم و سپس در هر لایه حاصل لایه قبل را با عنصر جدید xor میکنیم. به این روش تعداد لایه ها تتای n خواهد شد.

Y=(((x1 xor x2) xor x3) xor x4) xor .....

d) آیا می‌توان با یک شبکه عصبی MLP با یک لایه مخفی این کار را انجام داد؟ اگر بله حداقل تعداد نورون‌های لایه مخفی چقدر خواهد بود؟ (راهنمایی: از فرم sop[1] تابع منطقی XOR استفاده کنید)

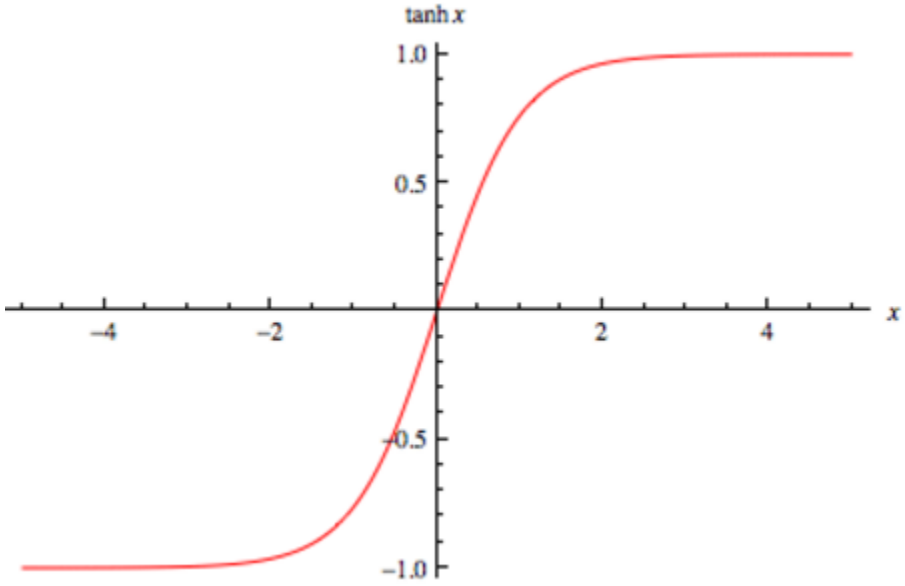
پاسخ : sop برای xor به ازای n ورودی ، 2^{n-1} ضرب دارد . بنابراین می بایست به همین تعداد نورون باشد تا تمام ورودی ها با وزن های مختلف که برابر وزن آنها در sop است به نورون ها متصل شوند و در نهایت خروجی نورون ها با وزن یک به لایه نهایی برود.

۲) فرض کنید مطابق شکل زیر یک شبکه عصبی MLP با چهار ورودی داریم. لایه پنهان اول آن به نام A دارای چهار نورون، لایه پنهان دوم آن به نام D دارای سه نورون و لایه سوم پنهان آن به نام E دارای دو نورون است. همچنین در لایه خروجی یک نورون به نام Y داریم. (دقت شود که خروجی نورون A2 علاوه بر لایه D به E1 نیز متصل است)
لازم به ذکر است که تابع فعال سازی در لایه‌های پنهان به ترتیب از نوع tanh و tanh و swish با پارامتر یک و در لایه خروجی از نوع sigmoid است.



a) با مقدار دهی اولیه بایاس ها به صفر و وزن ها به ۰٫۵ با ورودی‌های X0 = 1 و X1=1 و X2=1 و X3=1 مقدار خروجی شبکه عصبی را حساب کنید.

پاسخ: با استفاده از تابع فعالسازی tanh خروجی لایه A را محاسبه میکنیم:



A1= tanh(1 * .5 + 1 * .5 + 1 * .5 + 1 * .5) = 0.96

A2= tanh(1 * .5 + 1 * .5 + 1 * .5) = 0.90

A3= tanh(1 * .5 + 1 * .5 + 1 * .5) = 0.90

A4= tanh(1 * .5 + 1 * .5 + 1 * .5 + 1 * .5) = 0.96

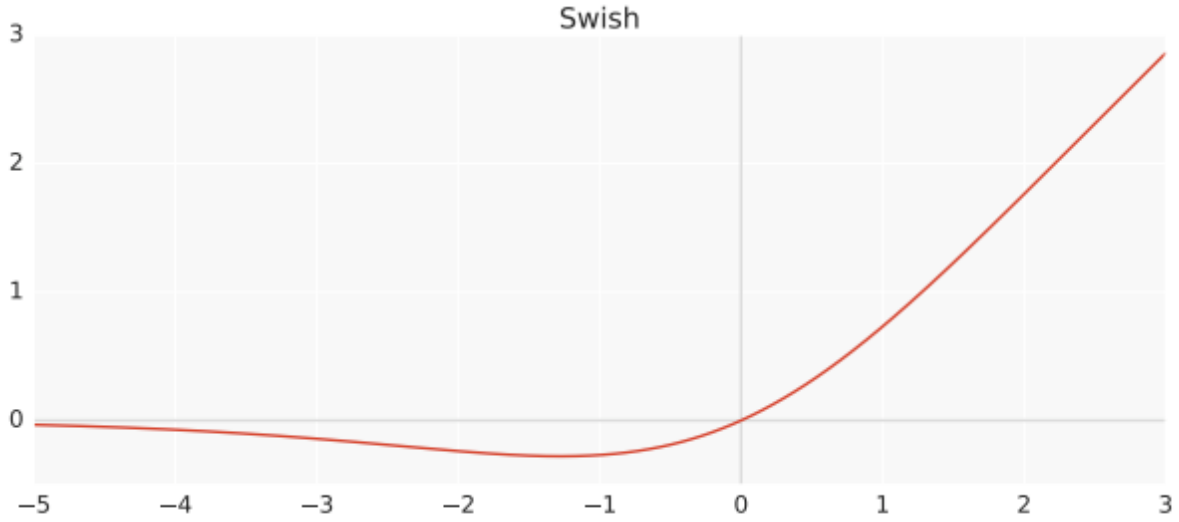
اکنون به سراغ محاسبه خروجی های لایه بعدی میرویم:

D1 = tanh(A1 *.5 + A2 * .5 + A4 * .5) = 0.88

D2 = tanh(A1 *.5 + A2 * .5 + A3 * .5 + A4 * .5) = 0.95

D2 = tanh(A1 *.5 + A2 * .5 + A3 * .5 + A4 * .5) = 0.95

تصویر زیر تابع فعالیت swish را با فرمول f(X)=x*sigmoid(bx) نشان میدهد.



بنابراین خروجی لایه E برابر است با:

E1= swish(D1 * 0.5 + D2 * 0.5 + D3 * .5 + A2 * 0.5) = swish(1.84)= 1.84*sigmoid(1.84)

= 1.84 * 0.86 = 1.58

E2= swish(D1 * 0.5 + D2 * 0.5 + D3 * .5)=swish(1.39)= 1.39 * 0.80 = 1.11

و در نهایت:

Y= sigmoid (E1 * 0.5 + E2 * 0.5) = sigmoid(1.34)=0.79

b) اگر خروجی مورد انتظار شبکه با ورودی های بالا برابر یک باشد با تابع هزینه کمترین مربعات خطا[۲] و با نرخ یادگیری[۳] 0.1 وزن های خروجی A2 را به روز رسانی کنید.

$$E = \frac{1}{2} * (Y - y)^2$$

⇨ E = ½ * $(1 - 0.79)^2$ = 0.02

اکنون می بایست میزان ارتباط خطا با خروجی های A2 را جهت به روزرسانی آنها به دست بیاوریم یا با زبان ریاضی از آنها گرادیان بگیریم.

در ادامه محاسبات فرآیند به دست آمدن گرادیان را مشاهده میکنید:

$$\frac{\partial E}{\partial A2} = \frac{\partial E}{\partial y\ out} * \frac{\partial y\ out}{\partial y\ in} * ((\frac{\partial y\ in}{\partial e1\ out} * \frac{\partial e1\ \text{out}}{\partial e1\ \text{in}} * (\frac{\partial e1\ in}{\partial d1} * \frac{\partial d1}{\partial a2} + \frac{\partial e1\ in}{\partial a2} + \frac{\partial e1\ in}{\partial d2} * \frac{\partial d2}{\partial a2} + \frac{\partial e1\ in}{\partial d3} * \frac{\partial d3}{\partial a2})) + (\frac{\partial y\ in}{\partial e2\ out} * \frac{\partial e2\ out}{\partial e2\ \text{in}} * (\frac{\partial e2\ in}{\partial d1} * \frac{\partial d1}{\partial a2} + \frac{\partial e2\ in}{\partial d2} * \frac{\partial d2}{\partial a2} + \frac{\partial e2\ in}{\partial d3} * \frac{\partial d3}{\partial a2}))$$

$$E = 1/2 * (Y - y\ out)^2 \Rightarrow \frac{\partial E}{\partial y\ out} = 2 * 1/2 * (Y - y) * -1 = y - Y = 0.79 - 1 = -0.21$$

$$y\ out = sigmoid(y\ in) \Rightarrow \frac{\partial y\ out}{\partial y\ in} = sigmoid(y\ in) * (1 - sigmoid(y\ in)) = 0.79 * 0.29 = 0.16$$

$$y\ in = 0.5 * e1\ out + 0.5 * e2\ out \Rightarrow \frac{\partial y\ in}{\partial e1\ out} = 0.5\ \&\ \frac{\partial y\ in}{\partial e2\ out} = 0.5$$

$$e1\ \ out = swish(e1\ in) \Rightarrow \frac{\sigma e1\ out}{\partial e1\ in} = swish(e1\ in) + sigmoid(e1\ in) * (1 - swish(e1\ in)) = 1.58 + 0.86 * 0.58 = 2.06$$

$$e2\ \ out = swish\left(e2\ in\right) \Rightarrow \frac{\partial e2\ out}{\partial e2\ in} = swish\left(e2\ in\right) + sigmoid\left(e2\ in\right) * \left(1 - swish(e2\ in)\right) = 1.11 + 0.80 * 0.11 = 1.19$$

$$e1\ in = 0.5 * (d1 + a2 + d2 + d3) \Rightarrow \frac{\partial e1\ in}{\partial d1} = \frac{\partial e1\ in}{\partial d2} = \frac{\partial e1\ in}{\partial d3} = \frac{\partial e1\ in}{\partial a2} = 0.5$$

$$e2\ in = 0.5 * \left(d1 + d2 + d3\right) \Rightarrow \frac{\partial e2\ in}{\partial d1} = \frac{\partial e2\ in}{\partial d2} = \frac{\partial e2\ in}{\partial d3} = 0.5$$

$$\frac{\partial d1}{\partial a2} = \frac{\partial d2}{\partial a2} = \frac{\partial d3}{\partial a2} = 0.5$$

$$\Rightarrow \boldsymbol{\frac{\partial E}{\partial a2}} = -0.05$$

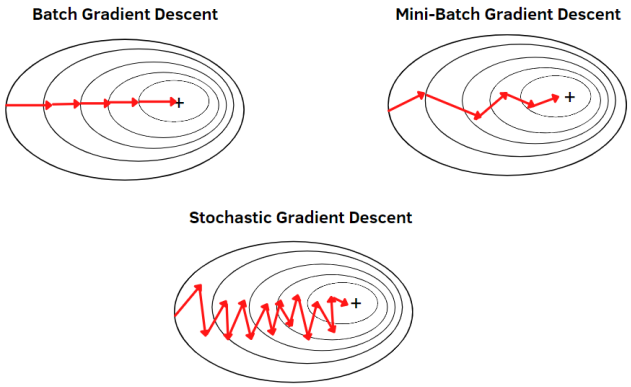
بنابراین گرادیان خطا بر a2 برابر است با 0.05 - و با توجه به نرخ یادگیری وزن به روزرسانی شده برابر است با:

$$w^+ = w + 0.1 * (-0.05) = 0.5 - 0.005 = 0.495$$

۳) به سوالات زیر پاسخ دهید.

a) تفاوت الگوریتم‌های stochastic gradient descent و batch gradient descent در چیست و هر کدام باید در چه مواردی استفاده شوند؟ فرق این دو الگوریتم با الگوریتم mini-batch gradient descent چیست؟

پاسخ: در الگوریتم batch ، ابتدا ارور را به ازای تمام داده های آموزشی محاسبه میکنیم و در آخر وزن های جدید را به دست میاوریم ولی در stochastic به ازای هر داده آموزشی وزن ها را آپدیت خواهیم کرد.
در الگوریتم batch اگر دیتاست ما بزرگ باشد میزان محاسبات ما خیلی زیاد خواهد شد در حالی که با استفاده از الگوریتم stochastic سریع تر و راحت تر میتوان وزن های یک دیتاست بزرگ را در حالت همگرایی پیدا کرد. الگوریتم mini – batch ترکیبی از دو الگوریتم قبل است که بعد از تعداد مشخصی از داده های آزمایشی تغییرات را اعمال خواهد کرد.



b) توضیح دهید با افزایش مقدار لامبدا (پارامتر منظم‌سازی[۴]) در عملکرد مدل چه تغییری حاصل شده و تاثیر آن در بیش‌برازش چیست؟ در چه مواقعی از منظم‌سازی L1 و در چه مواقعی از منظم‌سازی L2 استفاده میکنیم؟

در رگرسیون L1&L2 هدف این است که با تعریف یک هزینه یا جریمه ، مدلی را که به دست می آوریم خطای کمتری داشته باشد . این روش را میشود با مدل کردن با حساسیت کمتر به داده های آموزشی برآورده کرد. در این روش مجذور یا قدر مطلق شیب بین نقاط آزمایشی را (متناسب با نوع رگرسیون) در یک مقدار که پارامتر منظم سازی یا لامبدا نام دارد ضرب میکنیم که در واقع این یعنی پارامتر وابسته ما تا چه حد به تغییرات پارامترهای مستقل وابسته باشد. هر چه لامبدا بزرگتر باشد این وابستی بیشتر و احتمال خطا بیشتر میشود . از نقطه نظر عملی، L1 تمایل دارد ضرایب را به صفر کاهش دهد در حالی که L2 تمایل دارد ضرایب را به طور مساوی کاهش دهد. بنابراین L1 برای انتخاب ویژگی مفید است، زیرا می‌توانیم متغیرهای مرتبط با ضرایب را که به صفر می‌رسند حذف کنیم. از سوی دیگر، L2 زمانی مفید است که ویژگی های هم خطی/همبسته داشته باشید.

c) یکی از روش‌های جلوگیری از بیش‌برازش[5] روش Dropout است. در مورد این روش تحقیق کنید و نحوه عملکرد آن را توضیح دهید. اگر حذف برخی از نورون‌ها در این تکنیک منجر به عملکرد بهتر مدل می‌شود، چرا از اول از یک شبکه عصبی ساده‌تر با لایه‌های کمتر و نورون‌های کمتر استفاده نمی‌کنیم؟

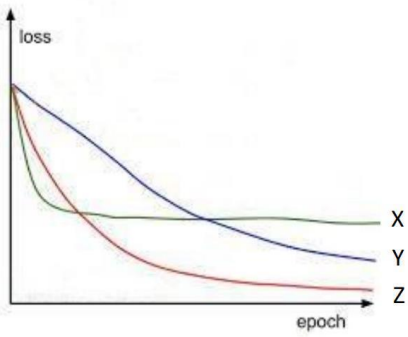
پاسخ:

در این روش زمانی که بیش برازش رخ داده است با یک مقدار تصادفی بین صفر و یک را به هر نورون اختصاص میدهیم. در این شرایط نورون هایی که مثلا از نیم کمتر است را حذف میکنیم. به این صورت مدل ما دیگر نتیجه ای فیت با داده های تست تولید نخواهد کرد. همچنین میبایست توجه داشت که نورون های لایه خروجی حذف نخواهند شد.

یکی از دلایلی که این روش ممکن است به ما کمک کند این است که بعضی وقت ، زمانی که ما مدل را آموزش داده ایم ، بعضی از نورون ها هستند که اشتباهات نورون های دیگر را پوشش دهند و عملا مقدار خطای ما صفر شود. در این حالت است که در دیگر داده ها که مدل آنها را ندیده است خطا پیش می آید. از این رو با استفاده از این روش برخی از نورون ها به صورت رندوم حذف میشوند که این مشکل بر طرف شود و این مجزا از این است که مدل را از نظر طراحی ساده کنیم .

d) نمودار خطا بر حسب شماره epoch زیر از آموزش یک شبکه عصبی یکسان با نرخ یادگیری های متفاوت با مقادیر X, Y, Z بدست آمده است. این مقادیر را با هم مقایسه کنید (رابطه هر سه مقدار را با هم توضیح دهید).

پاسخ:



در هر دور از محاسبه خطا و به روز رسانی وزن ها و بایاس که به آن ایپاک نیز میگوییم ، وقتی نرخ یادگیری را زیاد بکنیم خطا هم احتمالا بیشتر شود . بیشتر شدن خطا به این معنی است که احتمالا نوساناتی را حول مینیمم های محلی خواهیم داشت. اما وقتی نرخ یادگیری کمتر باشد این روند طولانی اما با احتمال خطای کمتر است . در نمودار بالا تغییر مقدار خطای ناگهانی نشان از همین نوسانات دارد. بنابراین شیب نمودار ها آنها تا زمان رسیدن به همگرایی نشان از نرخ یادگیری دارد که به این صورت x>z>y

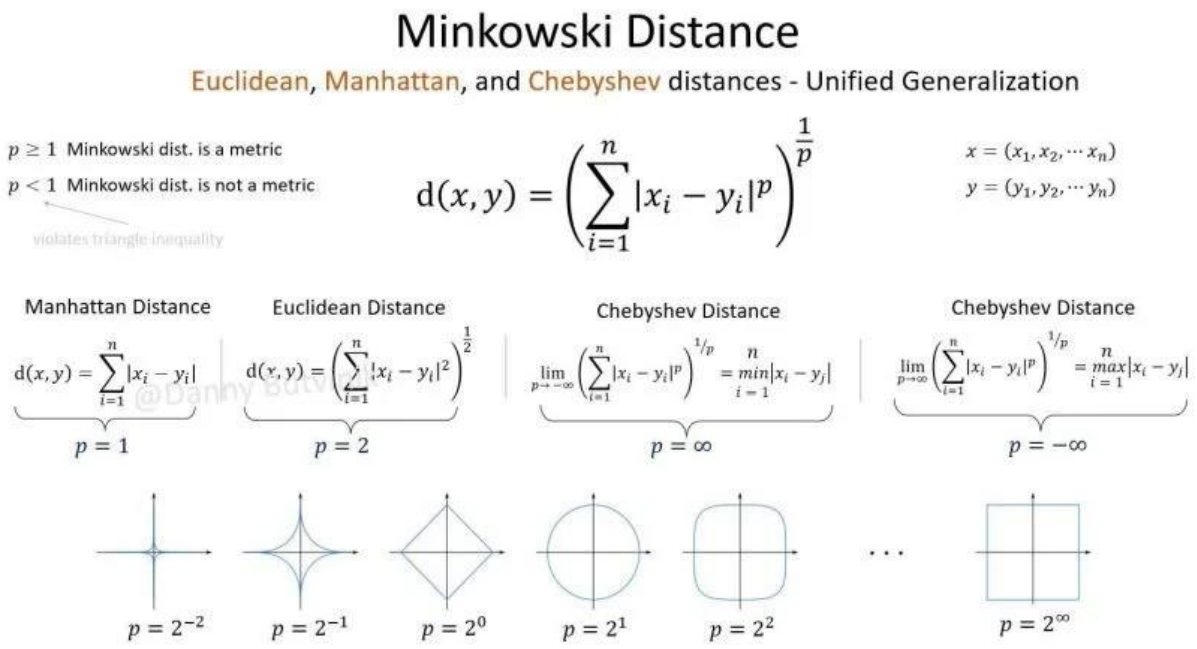
۴) همانطور که می‌دانید در شبکه‌های RBF وزن‌های وارد شونده به لایه میانی همان مراکز نواحی تحت پوشش شبکه هستند که فواصل ورودی‌های شبکه نسبت به آن‌ها محاسبه می‌شود و وارد تابع فعال سازی نورون‌های میانی می‌شود. نحوه محاسبه فاصله بین وزن‌ها و ورودی‌های شبکه شکل ناحیه تحت پوشش شبکه را مشخص می‌کند. یک خانواده معروف از توابع فاصله، خانواده مینکوفسکی[۶] است که به شکل زیر تعریف می‌شود: (x,y دو بردار هستند)

$$d_k(x,y) = (\sum_{i=1}^{n}(x_i - y_i)^k)^{\frac{1}{k}}$$

a) مشخص کنید که به ازای k=1 و k=2 این تابع برابر با کدام یک از توابع فاصله معروف می‌شود و شکل ناحیه تحت پوشش شبکه در این حالت چگونه خواهد بود.

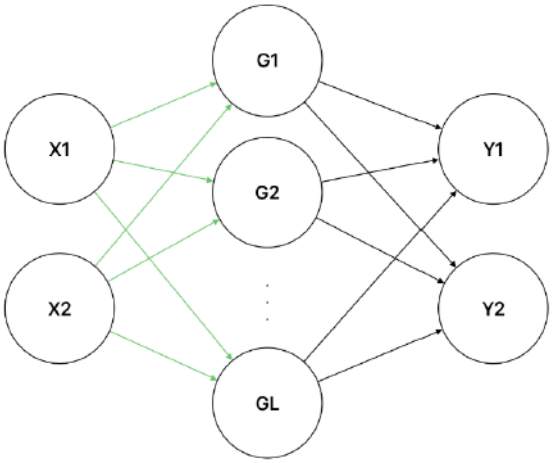
b) اگر k به بی‌نهایت میل کند تابع فاصله حاصل چه چیزی خواهد بود؟ شکل ناحیه تحت پوشش شبکه را در این حالت توصیف کنید.

پاسخ: در تصویر پایین میتوان حالات مختلف بابع مینکوفسکی را به ازای k های مختلف دید



بنابراین به ازای k=1 حاصل Manhattan distance با ناحیه تحت پوشش مربعی و به ازای k=2 حاصل Euclidean distance با ناحیه تحت پوشش دایره ای است. به ازای k مساوی بی نهایت نیز chebyshev distance خواهد بود که ناحیه تحت پوشش قابل ملاحظه است.

۵) شکل زیر یک شبکه RBF را نشان می‌دهد که دارای L نورون در لایه میانی است. فرض کنید وزن‌های ورودی به لایه میانی که همان مراکز شبکه هستند(رنگ سبز) از قبل مشخص شده اند.
اگر تعداد داده‌های آموزشی m باشد، ماتریس وزن‌های ورودی به لایه آخر شبکه(W) را برحسب ماتریس مقادیر لایه میانی شبکه(G) و ماتریس مقادیر خروجی مطلوب(Y) محاسبه کنید. در محاسبات خود از منظم‌سازی L2 استفاده کنید و ابعاد ماتریس‌ها را نیز ذکر کنید.



پاسخ:

ورودی به صورت یک ماتریس دو در یک است که وقتی بخواهیم آن را به یک ماتریس کلی از m داده آموزشی تبدیل کنیم یک ماتریس با ابعاد (2,m)خواهیم داشت. همچنین خروجی لایه میانی برابر است با مجموع exp(-landa * (x-w)^2) . همچنین ابعاد ماتریس نهایی برابر است با (1, L). همچنین از L2 میتوان در اضافه کردن مجذور لاندا به تابع هزینه استفاده کرد.

۶) به سوالات زیر پاسخ دهید.

a) خروجی لایه کانولوشنی زیر را حساب کنید.

Input

1	2	3	3
0	1	2	2
1	2	3	0
1	2	3	0

*

kernel

1	-1	1
-1	1	-1
1	-1	1

پاسخ:

با ضرب این فیلتر یک ماتریس دو در دو خواهیم داشت :

O =

3	0
1	1

b) اگر خروجی واقعی بالا ماتریس همانی باشد و بخواهیم با توجه با تابع هزینه کمترین مربعات خطا و نرخ یادگیری ۰,۱ وزن های کرنل خود رو به روز رسانی کنیم بعد از یک مرحله به چه وزن هایی می رسیم؟

پاسخ:

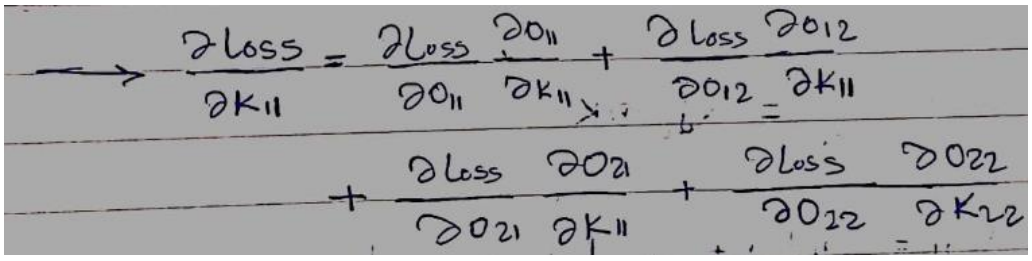
Y مقدار صحیح است . بنابراین خطا برابر است با :

$E = ¼ * ((o11-1)^2 + (o12-0)^2 + (o21-0)^2 + (o22-1)^2)$

O1= I 11 * K11 + I 12 * K 12 + I 13 * K 13 * + .../ O12 = ... , O21 = ... , O22 = ...

⇨ $K1\ new = K1 - 0.1 * (\frac{\partial E}{\partial K1})$

گرادیان برابر است با تصویر زیر:



$\frac{\partial E}{\partial O11}$ = ¼*(2*(O11-1))= ½*(2)=1 $\frac{\partial E}{\partial O12}$ = ¼*(2*(O12))=0 $\frac{\partial E}{\partial O21}$ = ¼*(2*(O21-0))= ½*(1)=1/2

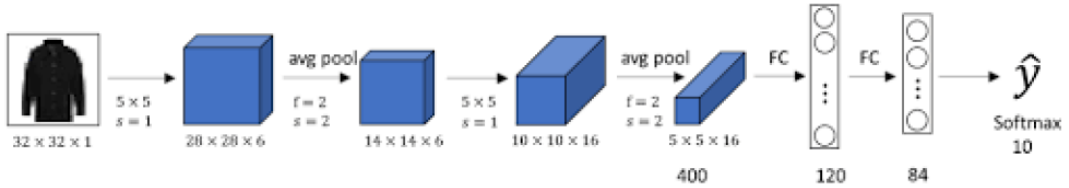
$\frac{\partial E}{\partial O22}$ = ¼*(2*(O22-1))= 0 $\frac{\partial 11}{\partial k11}$ = 1 $\frac{\partial 12}{\partial k11}$ = 2 $\frac{\partial 21}{\partial k11}$ = 0 $\frac{\partial 22}{\partial k11}$ = 1

⇨ **K11 new = k11 - .1 * (1+0+0+0)=0.9**

با استفاده از همین روش فیلتر جدید برابر است با:

0.9	-1.25	0.6
-1.1	0.8	-0.65
0.85	-0.7	0.7

c) فرض کنید برای پیاده‌سازی سه لایه آخر شبکه عصبی زیر به جای استفاده از لایه های تماما متصل[۷] از لایه های کانولوشنی استفاده کنیم پیشنهاد شما چیست ؟



راهنمایی: به ابعاد ورودی و خروجی توجه کنید.

پاسخ:

پس از آن که یک ماتریس 5*5*16 تحویل گرفتیم یک فیلتر 2*2*7 با s=1 قرار میدهیم تا حاصل بشود یک ماتریس 4*4*10. و در یک مرحله دیگر یک 4*4 max pool را اعمال میکنیم و یک وکتور ده تایی به دست می آید و سپس از طریق softmax می توان نتیجه را به دست آورد.

۷) مجموعه داده ای از ۵۰۰ نقطه را در یک فضای دوبعدی در نظر بگیرید، با مختصات x و y که به طور تصادفی از توزیع یکنواخت بین ۰ و ۱ تولید می شوند. یک SOM با یک شبکه 5x5 از نورون ها برای خوشه بندی این مجموعه داده طراحی کنید. وزن نورون ها را به طور تصادفی آغاز کنید. از یک توزیع یکنواخت بین ۰ و ۱، و از تابع همسایگی گاوسی با نرخ یادگیری رو به کاهش استفاده کنید. در مورد این شبکه به سوالات زیر پاسخ دهید.

a) مفهوم تابع همسایگی در SOM و چگونگی تأثیر آن بر فرآیند یادگیری را توضیح دهید.

پاسخ : تابع همسایگی ، نود هایی هستند به همسایگی نود برنده (درقبال ورودی) که با شعاع تعریف شده از قبل با ضریبی متناسب با فاصله تا نود برنده آپدیت خواهند شد. با این کار نود های همسایگی نسبتی از یک ویژگی را درون خود ایجاد خواهند کرد.

b) نقش نرخ یادگیری در SOM چیست و چگونه در طول آموزش به روز می شود؟

پاسخ: نرخ یادگیری میزان به روزرسانی نود ها را در قبال ویژگی های ورودی نشان میدهد که با پیشرفت روند آموزش این مقدار کاهش خواهد یافت

c) در مورد مقدار دهی اولیه وزن ها در SOM و اهمیت آن در فرآیند یادگیری توضیح دهید.

پاسخ: مقدار دهی اولیه ، یک مقدار دهی رندوم است که با پیشرفت روند آموزش به روز رسانی و مجموعه ای از همسایه ها با ویژگی های مشابه را خواهیم داشت

d) توضیح دهید که چگونه اندازه شبکه SOM بر عملکرد خوشه بندی و پیچیدگی محاسباتی تأثیر می گذارد.

پاسخ: وقتی نود برنده و همسایه ها بر اساس ورودی به روزرسانی شوند ، این روند در دراز مدت خوشه هایی را به وجود خواهند آورد با ویژگی های نسبی نه مطلق. که این روال عمل خوشه بندی را انجام میدهد.

(e

ابتدا برای ورودی x1 فاصله اقلیدسی نسبت به نود ها برابر است با :

Dis(X1 , node1) = 0.27

Dis(X1 , node2) = 0.29

Dis(X1 , node3) = 0.39