

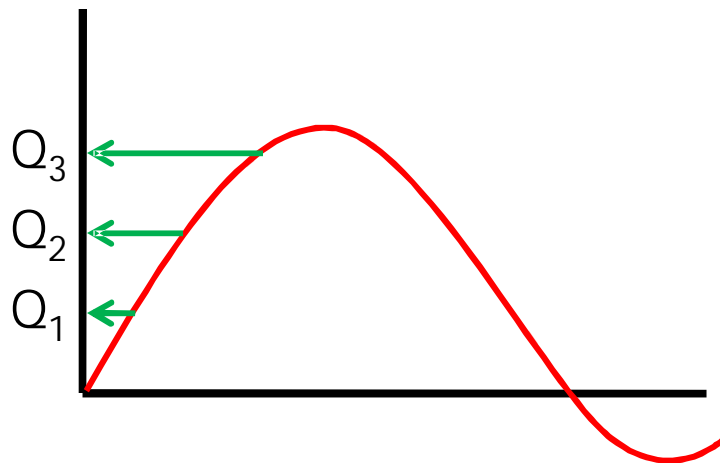
شبکه یادگیری کوانتیزاسیون برداری

Learning Vector Quantization Network

کوانتیزاسیون

۱۱ اسکالر

- کوانتیزاسیون یک بعدی
- مانند مبدل آنالوگ به دیجیتال (با داشتن 8 بیت ، دامنه موج آنالوگ به 256 سطح کوانتیزه میشود)



۱۲ برداری

- گسترش حالت فوق به چند بعد
- بردارها کوانتیزه میشوند

شبکه یادگیری کوانتیزاسیون برداری

ان کوانتیزاسیون برداری روشی است برای بهره برداری از ساختار موجود در بردار وزن ها.

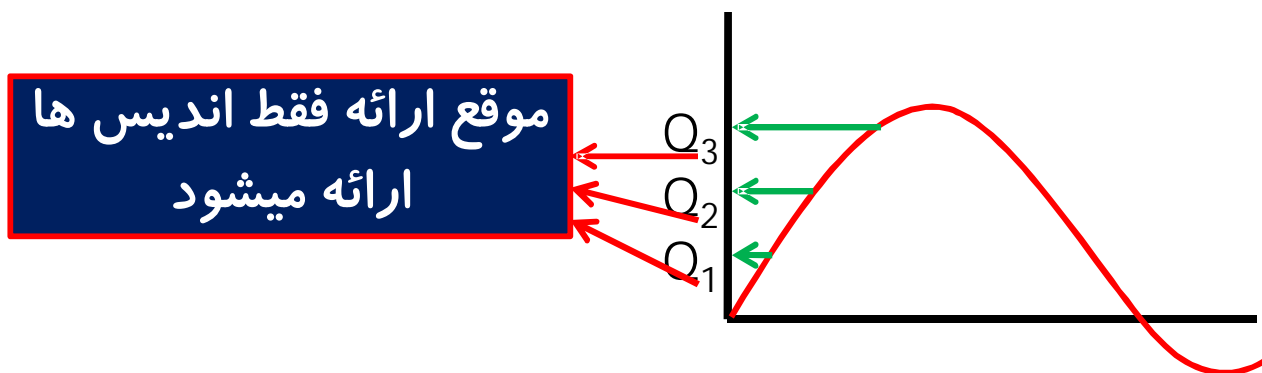
– به منظور فشردگی سازی داده ها

ان فضای ورودی به تعدادی ناحیه تقسیم می شود.

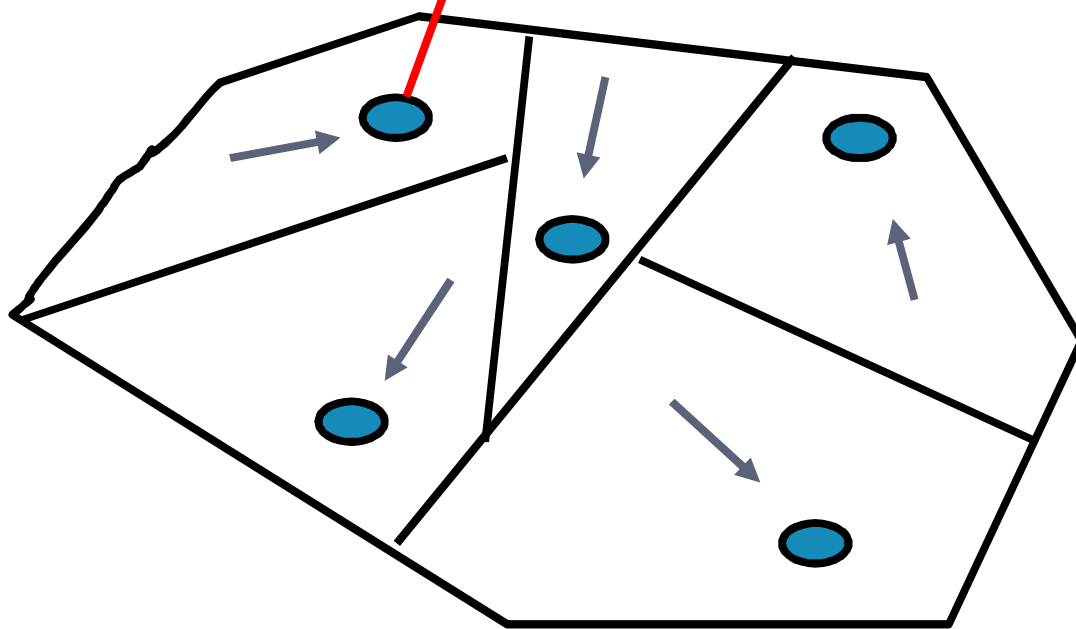
- برای هر ناحیه یک بردار بازسازی معین می شود.
- وقتی یک بردار جدید وارد سیستم می شود ابتدا ناحیه ای که بردار در آن واقع می شود تعیین شده و سپس بردار مذکور توسط بردار بازسازی آن ناحیه ارائه میشود.

۱۱ در این روش تعداد بیت های مورد نیاز به شدت کاهش پیدا میکند
- چون فقط شماره مربوط به بردار بازسازی بکار گرفته میشود.

۱۲ مجموعه تمام بردارهای بازسازی کتاب کد بازسازی نامیده می شود.
- در ارائه با بردارهای بازسازی تنها اندیس بردار بکار گرفته میشود
- مثال :



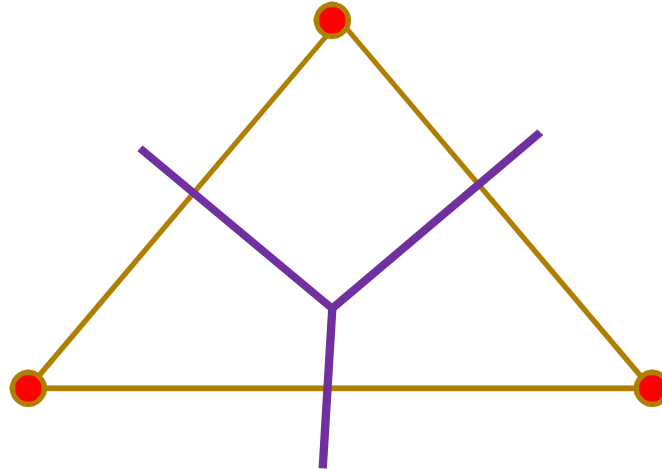
بردارهای بازسازی



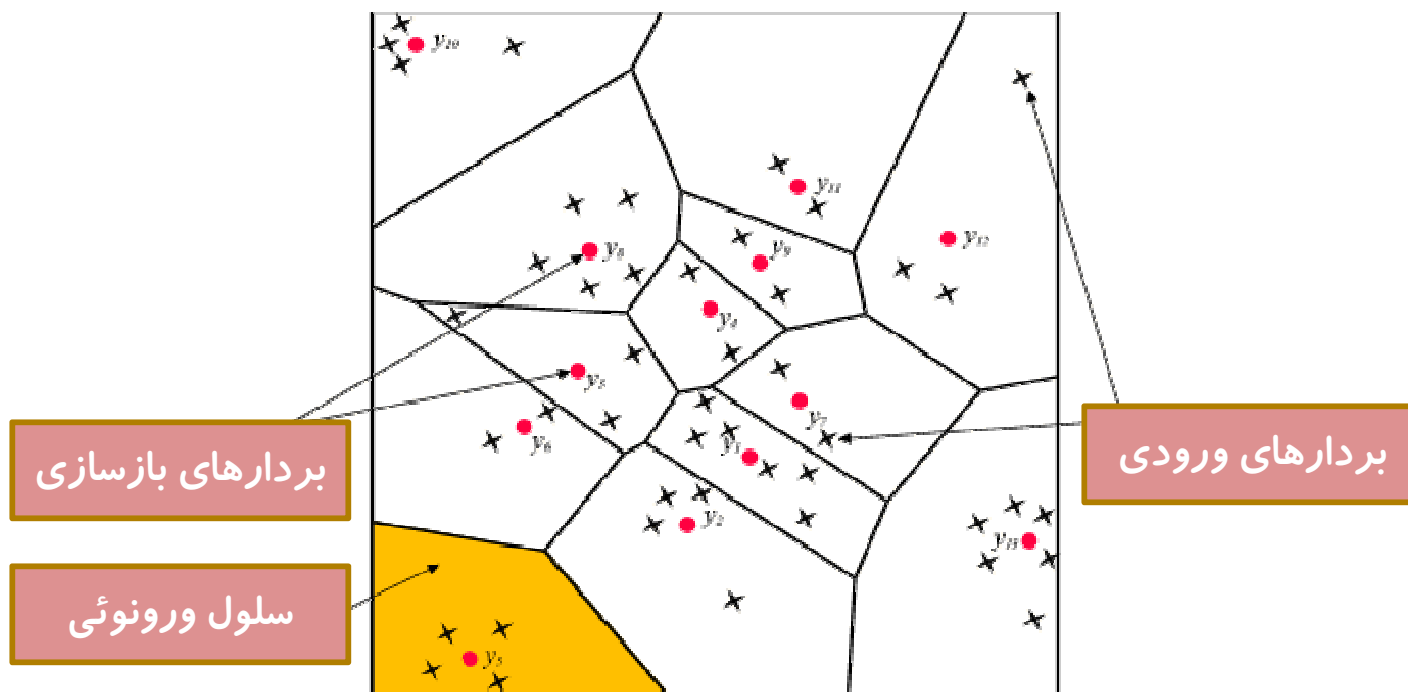
کوانتیزه کننده ورونوئی

- در یک کوانتیزه کننده برداری با حداقل خطا، تقسیم بندی فضا با استفاده از قاعده نزدیکترین همسایه مبتنی بر **فاصله اقلیدسی** است.

- به این کوانیزه کننده، **کوانتیزه کننده ورونوئی** می گویند.
- هر ناحیه را یک **سلول ورونوئی** می گویند.
- برای تشکیل این نواحی عمود منصف هر دو بردار رسم میشود



کوانتیزه کننده ورونوی



توضیحات بیشتر

- تعیین بردارهای ورونوئی میتواند به صورت تقریبی با یادگیری رقابتی انجام شود.

– بصورت بی نظارت

- برای دسته بندی الگوها از یادگیری کوانتیزاسیون برداری استفاده شده است.

– بصورت با نظارت

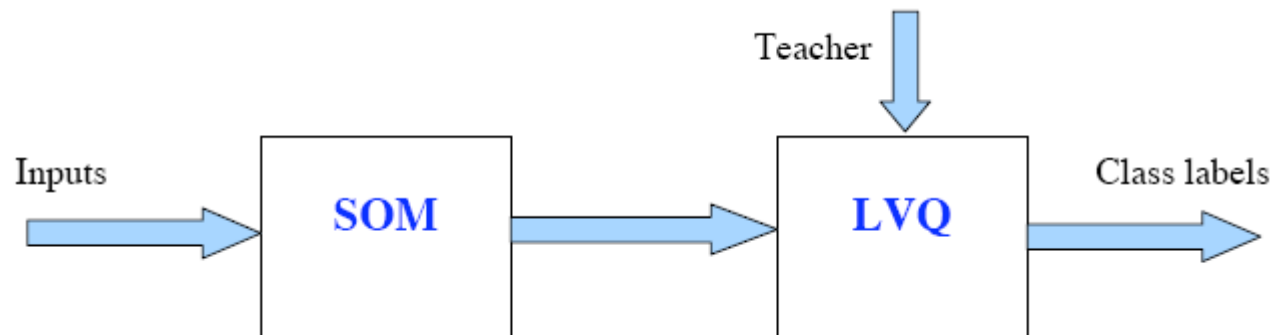
– اطلاعات مربوط به دسته های ورودی بکار گرفته میشود

– بردارهای ورونوئی که بروش بی نظارت تعیین شده بطور جزئی تغییر میکند.

- بطور کلی :

- تعیین بردارهای ورونوئی بصورت تقریبی میتواند توسط شبکه SOM که بی نظارت است مشخص شود.

- سپس میتوان برای تنظیم دقیق آنها از یک تکنیک دارای نظارت استفاده کرد.



LVQ چگونه عمل میکند ؟

- یک بردار ورودی \underline{S} به صورت اتفاقی از فضای ورودی انتخاب می شود:

– اگر دسته مربوط به \underline{S} با دسته مربوط به نزدیکترین بردار ورونوئی \underline{W} با هم تطبیق کنند، بردار \underline{W} به سمت \underline{S} جذب می شود.

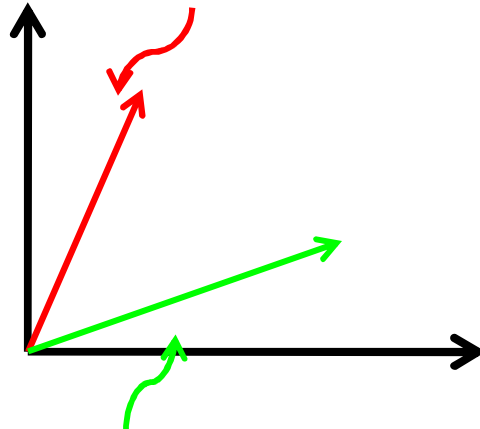
– اگر با هم تطبیق نکنند، \underline{W} مذکور از \underline{S} دور می شود.

• توضیح نمایشی

اگر دسته \underline{S} با دسته مربوط به \underline{W} با هم تطبیق کنند،
بردار \underline{W} به سمت \underline{S} جذب می شود.

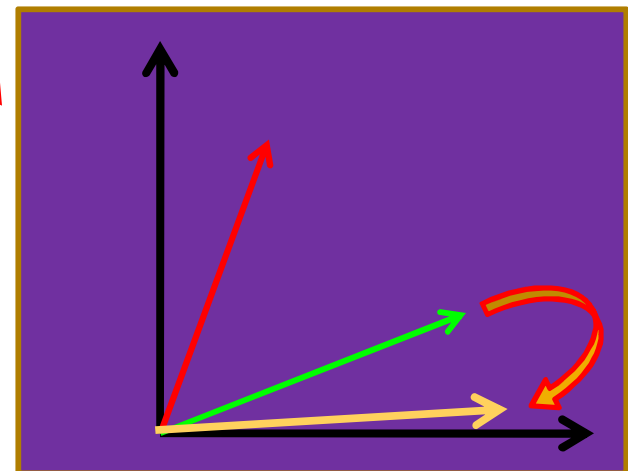
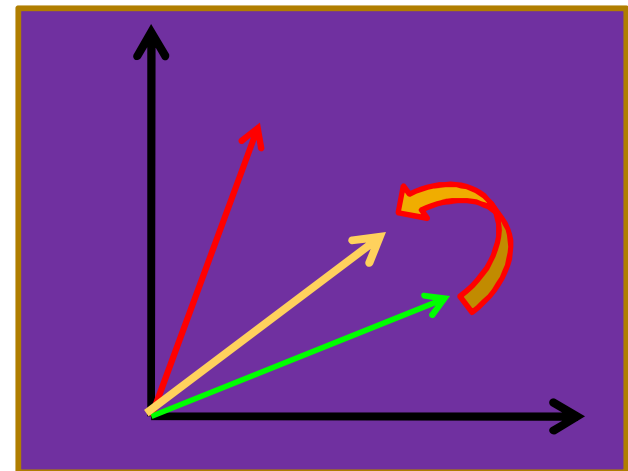
اگر دسته \underline{S} با دسته مربوط به \underline{W} با هم تطبیق نکنند،
بردار \underline{W} از \underline{S} دور می شود.

بردار ورودی \underline{S}



(نزدکترین بردار ورودی)

$\underline{W} =$ (به بردار ورودی)



فرض های الگوریتم LVQ

• فرض های در نظر گرفته شده:

– $\underline{S} = (S_1 \ S_2 \ \dots \ S_n)$ یک بردار آموزشی

– C_s دسته صحیح برای بردار \underline{S}

– \underline{W}_j بردار وزن واحد خروجی j ام

– C_j دسته ارائه شده با واحد خروجی j ام

– $||S - W_j||$ فاصله اقلیدسی بین بردار S و بردار وزن واحد

خروجی j ام

الگوریتم LVQ

0- بردارهای وزن واحدها مشخص شود.

مقدار اولیه α مشخص شود.

1- تا زمانی که شرط خاتمه غلط است قدم های 2 الی 6 تکرار شود.

2- برای هر بردار آموزشی S قدم های 3 و 4 انجام شود.

3- مقدار z که برای آن $||\underline{S} - \underline{W}_j||$ حداقل است مشخص شود.

4- مقدار W_j به صورت زیر بهنگام درآورده شود:

اگر $C_s = C_j$

$$\underline{W}_j(\text{new}) = \underline{W}_j(\text{old}) + \alpha (\underline{S} - \underline{W}_j(\text{old}))$$

اگر $C_s <> C_j$

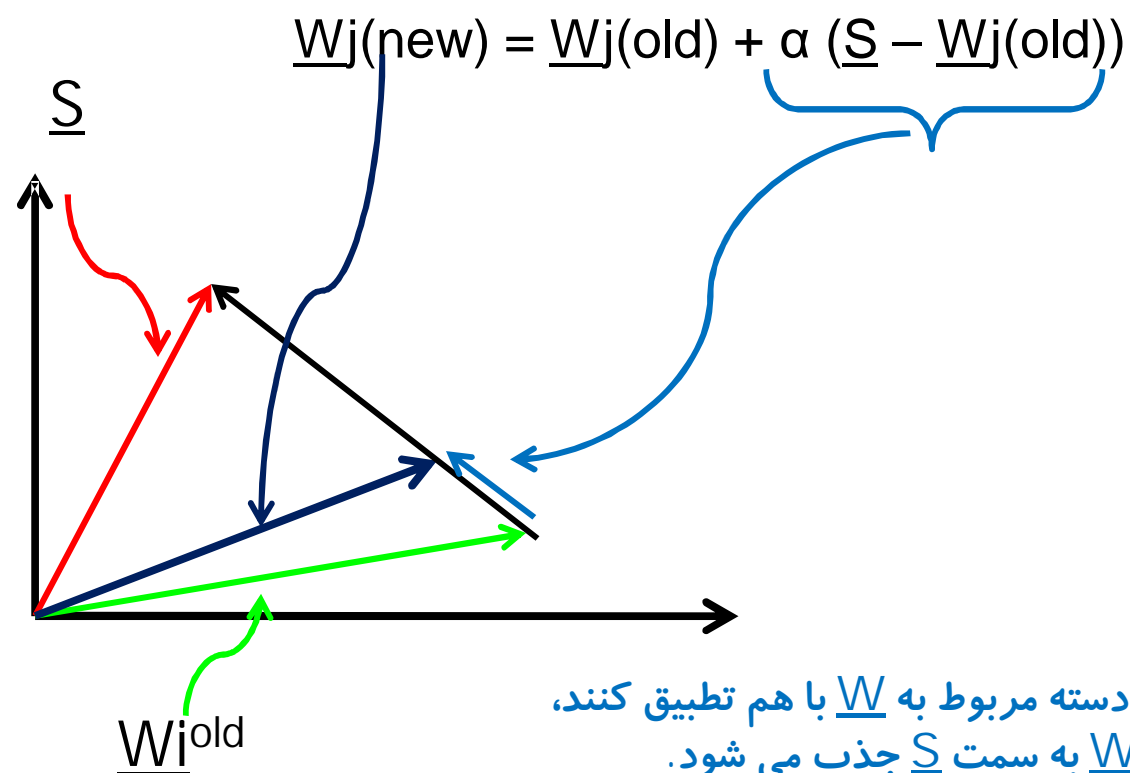
$$\underline{W}_j(\text{new}) = \underline{W}_j(\text{old}) - \alpha (\underline{S} - \underline{W}_j(\text{old}))$$

5- نرخ یادگیری α را کاهش دهید.

6- شرط خاتمه بررسی شود.

توضیح نمایشی مرحله 4

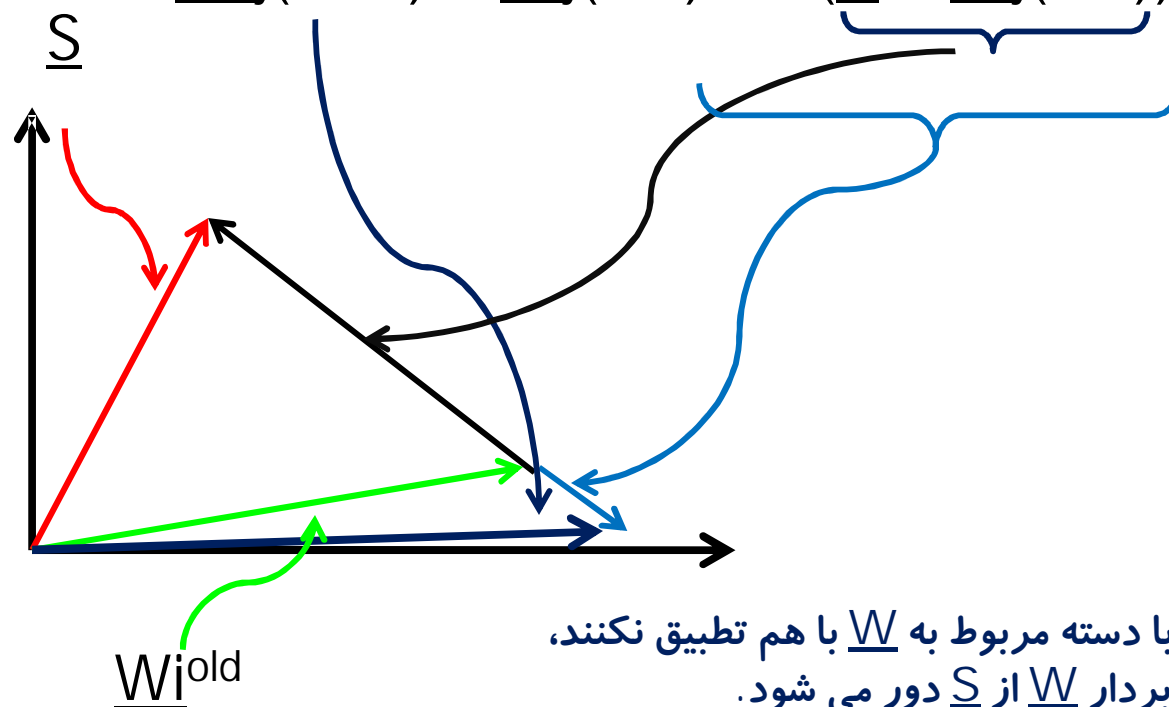
اگر $C_{\underline{S}} = C_j$:



توضیح نمایشی مرحله 4

– اگر $C_s \neq C_j$

$$\underline{W_j}(\text{new}) = \underline{W_j}(\text{old}) - \alpha (\underline{S} - \underline{W_j}(\text{old}))$$



اگر دسته \underline{S} با دسته مربوط به \underline{W} با هم تطبیق نکنند،
بردار \underline{W} از \underline{S} دور می شود.

تعیین اولیه وزن ها

- توسط شبکه SOM
- یک روش خوشه بندی، مانند k-means
- اولین m بردار آموزشی
- انتخاب m بردار آموزشی به عنوان مراکز

نسخ دیگر الگوریتم

- در نسخه شرح داده شده، تنها بردار برنده که نزدیکترین بردار به بردار ورودی است اصلاح می گردد.
 - در الگوریتم های اصلاح شده در هر قدم **دو بردار اصلاح می گردد که :**
 - یکی، نزدیکترین بردار به ورودی (بردار برنده)
 - و دیگری، نزدیکترین بردار بعدی به ورودی است (بردار نائب برنده)
- به شرط آنکه شرایط خاصی ارضاء گردد.**

شرایط خاص چیست ؟

- در صورت برقراری شرایط زیر، دو بردار (برنده و نائب برنده) اصلاح میشوند (نسخه LVQ2):

- بردار برنده و نایب برنده به دو دسته مختلف متعلق باشند.
- بردار ورودی متعلق به دسته بردار نایب برنده باشد.
- فاصله بردار ورودی تا بردار برنده و بردار نایب برنده تقریباً مساوی باشد.

نسخه LVQ2

- اگر \underline{S} بردار ورودی فعلی و \underline{W}_c نزدیکترین بردار وزن به \underline{S} و \underline{W}_r نزدیکترین بردار وزن بعدی به \underline{S} باشد و d_c و d_r به ترتیب فاصله بین بردارهای یاد شده باشد ($d_c < d_r$)

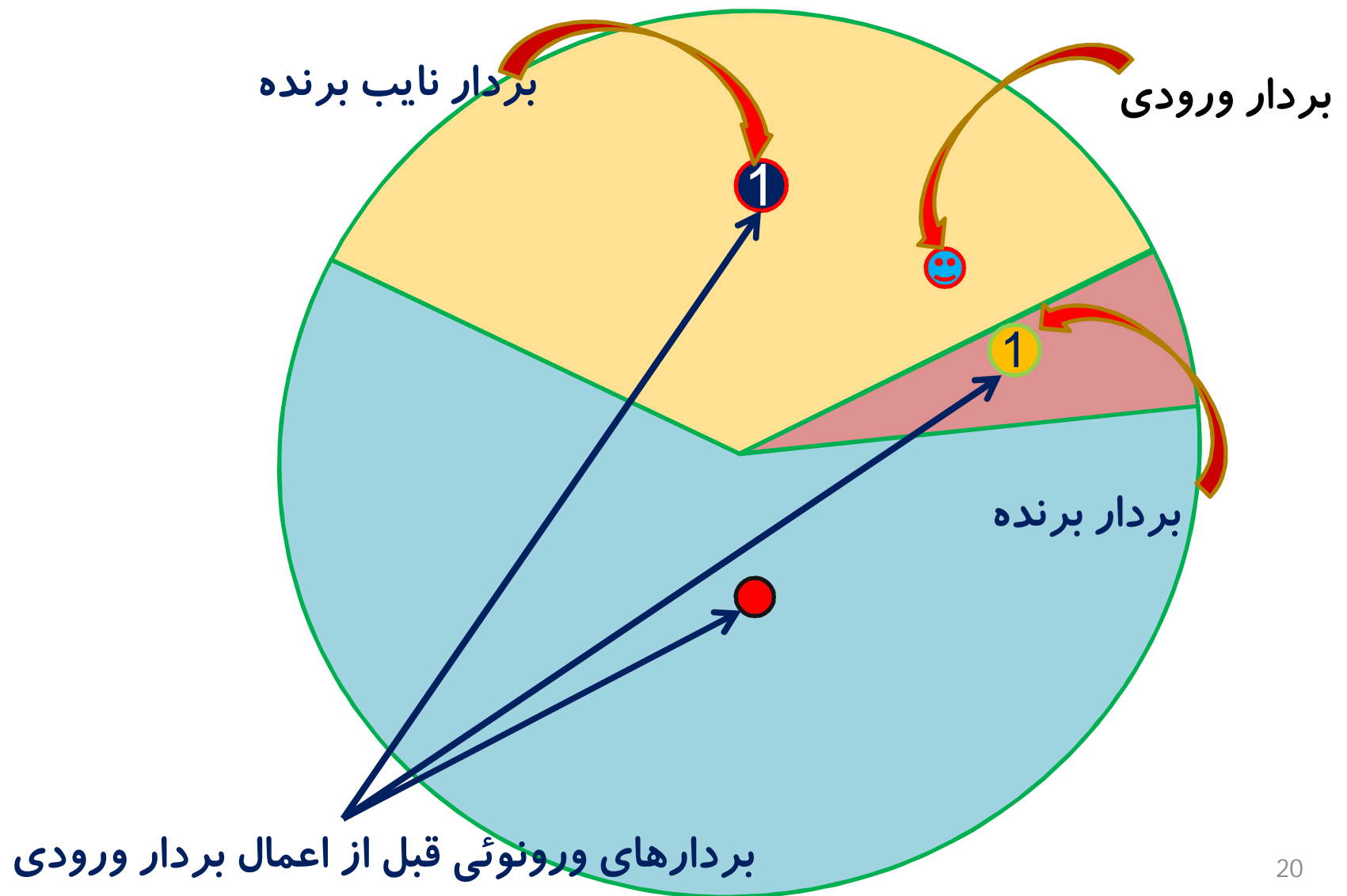
$$d_c/d_r > 1 - \epsilon \quad d_r/d_c < 1 + \epsilon$$

- در این الگوریتم، مشروط بر آنکه شرط فوق بین فواصل برقرار باشد و \underline{W}_c و \underline{W}_r متعلق به دو دسته مختلف باشد و \underline{S} متعلق به دسته \underline{W}_r باشد، \underline{W}_c و \underline{W}_r اصلاح می گردند.

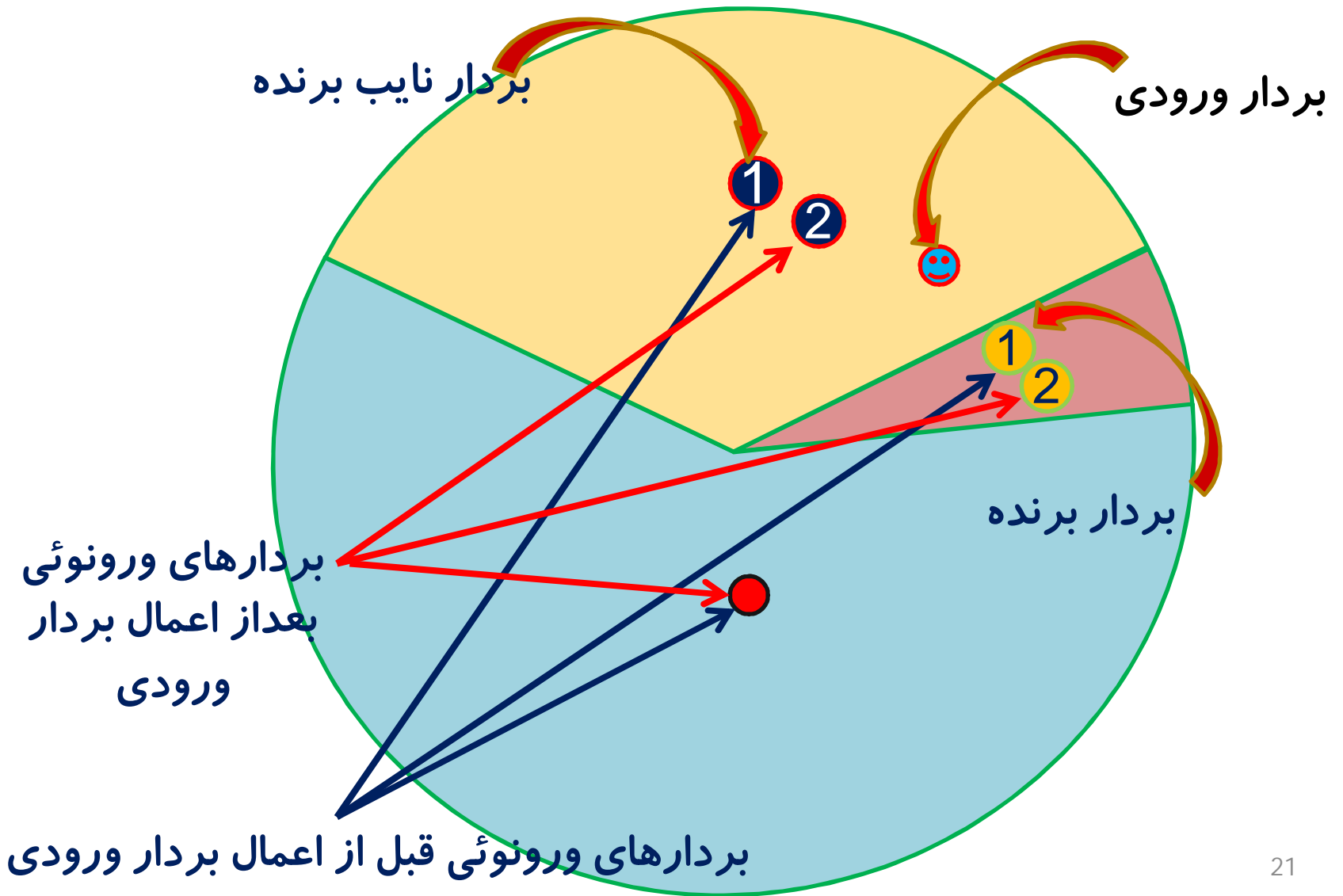
$$\underline{W}_c(t+1) = \underline{W}_c(t) - \alpha(t) (\underline{S}(t) - \underline{W}_c(t))$$

$$\underline{W}_r(t+1) = \underline{W}_r(t) + \alpha(t) (\underline{S}(t) - \underline{W}_r(t))$$

توضیح نمایشی نسخه LVQ2



توضیح نمایشی نسخه LVQ2



نسخه LVQ2.1

- در این نسخه دو بردار نزدیکتر به ورودی در نظر گرفته می شود ($\underline{Wc1}$ و $\underline{Wc2}$).

– این بردارها به شرطی اصلاح می شوند که یکی از آنها متعلق به کلاس صحیح برای ورودی فعلی \underline{S} باشد و دیگری متعلق به آن کلاس نباشد.

$$\min [dc1/dc2, dc2/dc1] > 1 - \epsilon$$
$$\max [dc1/dc2, dc2/dc1] < 1 + \epsilon$$

– در این حال برای بردار متعلق به کلاس ورودی \underline{S} :

$$\underline{Wc1}(t+1) = \underline{Wc1}(t) + \alpha(t) (\underline{S}(t) - \underline{Wc1}(t))$$

– و برای برداری که متعلق به کلاس نیست:

$$\underline{Wc2}(t+1) = \underline{Wc2}(t) - \alpha(t) (\underline{S}(t) - \underline{Wc2}(t))$$

• برای دسته بندی الگوها از یادگیری کوانتیزاسیون برداری استفاده شده است.

– بصورت با نظارت

– اطلاعات مربوط به دسته های ورودی بکار گرفته میشود

– بردارهای ورونوئی که بروش بی نظارت تعیین شده بطور جزئی تغییر میکند.

کوانتیزه کننده برداری سلسله مراتبی

- می توان همیشه با کد کردن بردار به جای کد کردن اسکالر، فشردگی سازی بهتری بدست آورد.
- اما کوانتیزاسیون برداری با حجم محاسبات بالا روبه روست.
- در کوانتیزه کننده سلسله مراتبی با فدا کردن کمی صحت، سرعت بطور قابل ملاحظه ای افزایش می یابد.

کوانتیزه کننده برداری سلسله مراتبی

