امیرحسن احمدی، ۹۷۵۲۲۲۹۲، گذارش تمرین دوم هوش محاسباتی.

Q1_A

یک کلاس Kohonen درست کردم و ابتدا n (اندازه ضلع فضای دو بعدی خروجی) را به کلاس میدهیم و یک سری مقدار پیشفرض برای learning_rate، sigma و تعداد ایتریشن ها در نظر میگیریم و بردار وزن ها را بر اساس n و سه مولفه بودن رنگ(rgb) به صورت رندوم میسازیم.

بعد تعدادی training_data (طبق صورت سوال ۱۶۰۰ تا) به صورت رندوم جنریت کرده و به کلاس میدهیم تا train شود.

در هر مرحله از train فاصله رنگ ورودی را از وزن ها حساب کرده و وزن برنده(کمترین فاصله) را پیدا میکنیم. بعد فاصله هر خانه از مپ خروجی از وزن برنده را محاسبه کرده و براساس فرمول h رنگ های در شعاع وزن برنده را آپدیت میکنیم و این کار را برای همه ی training_data ها به تعداد ایتریشن ها انجام میدهیم و در آخر تصویر وزن ها را خروجی میدهیم.

Q1_B

اگر learning_rate را تغییر ندهیم نمیتوانیم مطمعن باشیم که شبکه ما همگرا میشود و همیشه وزن تعداد زیادی نرون تغییر میکند که به آن Map distortions میگویند.

به همین دلیل بعد از هر ایتریشن learning_rate را در 9. ضرب میکنیم که کاهش یابد و چون برای مثال قبل learning_rate را کم گرفته بودیم که به یک جواب مطلوب برسیم، این بار learning_rate را از ۱ شروع میکنیم و مثل قسمت A عمل میکنیم.

Q1_C

مانند learning_rate اگر سیگما هم ثابت در نظر بگیریم و در مقدارش دقت نکنیم، اگر خیلی بزرگ باشد بیشتر صفحه را یک رنگ و اگر زیادی کوچک باشد از یک رنگ چندین منطقه به وجود میاید. پس بهتر است مانند learning_rate، سیگما را نیز در هر ایتریشن کاهش دهیم تا نتیجه ی بهتری بگیریم.

Q2_A

ابتدا ۳۰۰ عدد رندوم تولید کرده و به بازه 3 تا 3- مپ میکنیم و تابع سینوس را برای آن محاسبه میکنیم.

سپس مدل را ساخته و سه لایه آن را مانند سوال هفته قبل dence قرار میدهیم که فولی کانکتد است و چون مساله باینری کلسیفیکیشن نیست بهتر است از اکتیویشن فانکشن sigmoid استفاده کنیم.

سپس مدل را با دیتاهای آماده شده train میکنیم.(تعداد ایتریشن ها به صورت دستی و با آزمون خطا به دست آمده)

در آخر هم تعدادی دیتای تستی تولید کرده و برای prediction به مدل میدهیم و نمودار سینوس و دیتای predict شده را با plot رسم میکنیم.

Q2_B

مانند قسمت قبل دیتای train و test جدا ساخته و به ترتیب به تابع fit و predict میدهیم. در fit مراکز را به صورت رندوم انتخاب کرده فواصل تا مراکز را بدست آورده و با جواب ها دات میکنیم و وزن ها بدست میایند. در آخر نیز دو نمودار را چاپ میکنیم.

Q2_C

Rbf بسیار سریع تر است ولی در mlp با بیشتر کردن ایتریشن ها میتوان دقت را بالا تر برد.