

به نام خدا

نام و نام خانوادگی : حسین سیم چی

شماره دانشجویی : ۹۸۴۴۳۱۱۹

درس : یادگیری ماشین (تمرین اول)

عنوان : توضیح شکل ها و تحلیل کارهای انجام شده

زمستان و بهار ۹۸

فهرست

۳.....	مقدمه
۴.....	گزارش تحلیل کد
۴.....	توضیحات دیتاست
۵.....	معادله ی بسته
۵.....	گرادیان کاهشی
۶.....	تحلیل دیتاست خط (Line)
۸.....	نرخ یادگیری (۰/۰۰۱)
۱۷.....	نرخ یادگیری (۰/۰۱)
۲۲.....	نرخ یادگیری (۰/۲)
۲۹.....	نرخ یادگیری (۰/۵)
۳۵.....	نرخ یادگیری (۰/۹)
۴۰.....	جمع بندی

مقدمه :

در تمرین اول ، به بررسی رگرسیون خطی می پردازیم . باتوجه به صورت سوال که از چندین بخش تشکیل شده است در این فایل به بررسی شکلها و نتایج بدست آمده و تحلیل چگونگی علل رخداد این نمودارها و تصاویر می پردازیم .

گزارش کار شامل دو بخش میباشد ، بخش اول که همین فایل میباشد و شامل تحلیل شکل ها و تحلیل کارهای انجام شده میباشد و بخش دوم شامل تحلیل کد نوشته شده به صورت خط به خط به شکل فایل مقاله و به صورت انگلیسی در محیط Latex میباشد که در بخش مربوط به گزارش کار نیز قرار داده شده است .

باتوجه به ذکر این نکته در فایل تصویری کلاس و توضیحات آقای دکتر آبین ، استفاده از کتابخانه های آماده نظیر Sklearn برای تولید داده و انجام متدهای مختلف ممنوع میباشد ، لذا در کد نوشته شده که در فایل دوم به طور کامل توضیح داده شده است تنها از کتابخانه ی NumPy برای تولید داده استفاده شده است و سایر خط های کد نیز به صورت کاملا دستی نوشته شده است

گزارش تحلیل کد :

در پوشه ی مربوط به فایل گزارش ، در کنار همین فایل ، فایل دیگری نیز به صورت PDF قرار داده شده است که به فرمت مقاله و به صورت انگلیسی ، خط به خط کد نوشته شده را توضیح داده ام . در محیط Latex به دلیل وجود محیط های مختلف برای ریاضی و غیره صلاح را بر این دانستم که فایل کد که تقریباً فایلی شامل معادلات ریاضی و فرمولهای استفاده شده است را به این شکل بنویسم .

توضیحات دیتاست :

کد نوشته شده که در فایل های جداگانه قرار داده شده است ، شامل ۱۶ برنامه ی مختلف میباشد ، که تنها در قسمت تولید داده با یکدیگر تفاوت دارند و در قسمتهای بعدی نظیر محاسبه ی معادله ی بسته (Closed equation) یا محاسبه ی گرادیان محلی (Gradient descent) یکسان است لذا از توضیحات تکراری خودداری می کنم و صرفاً روند حرکت کار روی یک دیتاست خاص را توضیح میدهم . در فایل های ارسال شده ، از نرخ های یادگیری متفاوت استفاده شده است که از ۰/۰۰۱ شروع کرده و تا ۰/۹ آن را ادامه داده ام لذا در ادامه خواهیم دید که سرعت حرکت گرادیان به سمت مینیمم محلی چقدر به میزان نرخ یادگیری وابسته است .

معادله ی بسته (Closed equation) :

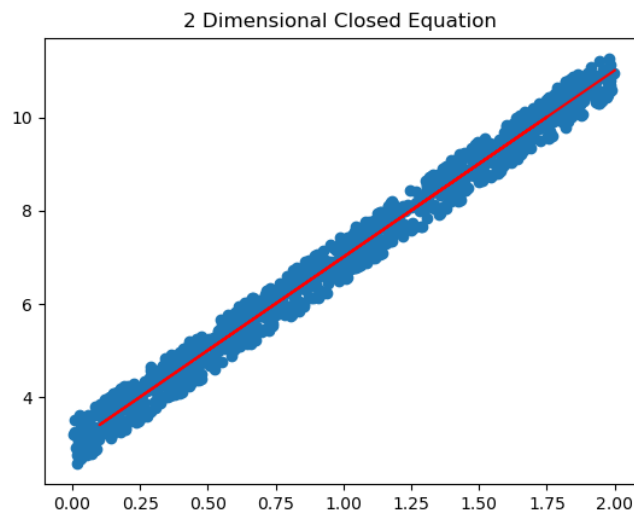
با صفر قرار دادن معادله ی مربوط به خط $(y = mx + c)$ ، به راحتی میتوان ضراب معادله را بدست آورد و سپس میتوان خط بهینه را به دیتا فیت کرد. خطی را خط بهینه می نامیم که کمترین خطا و بهترین کلاس بندی را برای داده ها داشته باشد (کد مربوط به این قسمت و نحوه ی محاسبه و معادله ی بسته به طور کامل شرح داده شده است).

گرادیان کاهشی (Gradient Descent Method) :

در این روش با درنظر گرفتن معادله ی خط انتظار داریم در هربار اپدیت ضرایب ، خطای این خط کم شده و به سمت مینیمم محلی همگرا شویم سرعت همگرایی در این روش کاملاً وابسته به نرخ یادگیری میباشد به گونه ایی که در ادامه خواهیم دید اگر میزان نرخ یادگیری را اعداد بالا قرار دهیم همگرایی در بیشتر اوقات رخ نخواهد داد .

تحلیل دیتاست خط (Line) :

در ابتدا با بررسی معادله ی بسته که در قسمت قبل توضیح داده شد خط مورد نظر را به داده های خود فیت می کنیم . همانطور که در توضیحات فایل کد ذکر شده است باتوجه به مختصات مکانی (X) که داریم و بدست آوردن زاویه مورد نظر در هربار به ازای تمامی سمپل های تعریف شده مقدار γ آنها را پیش بینی می کنیم که در واقع باید این خط ، خطی باشد که کمترین خطا را داشته باشد . کمترین خطا بدین معنی است که تفاضل میزان γ پیش بینی شده از γ واقعی نزدیک به صفر باشد . (نحوه ی بدست آوردن زاویه و تعریف سمپل ها به طور کامل در فایل توضیحات کد آمده است)



همانطور که در بالا قابل مشاهده است ، خط قرمز رنگ به درستی توانسته این تفکیک پذیری را انجام دهد

در ادامه به بررسی الگوریتم گرادیان محلی می پردازیم ، در این الگوریتم در ابتدا ضرایب معادله را صفر در نظر گرفته ایم و در ادامه انتظار داریم در هر بار اپدیت وزن این خط به سمت خط بهینه با کمترین خطا همگرا شود .

زمانی که ضرایب را در ابتدا صفر قرار می دهیم همانطور که در شکل پایین نیز قابل مشاهده است در اولین اپوک خط ما مطابق انتظار روی محور x ها مماس خواهد بود ($y = 0$) و در اپدیت های بعدی این خط به مرور به خط بهینه ی ما نزدیک می شود .

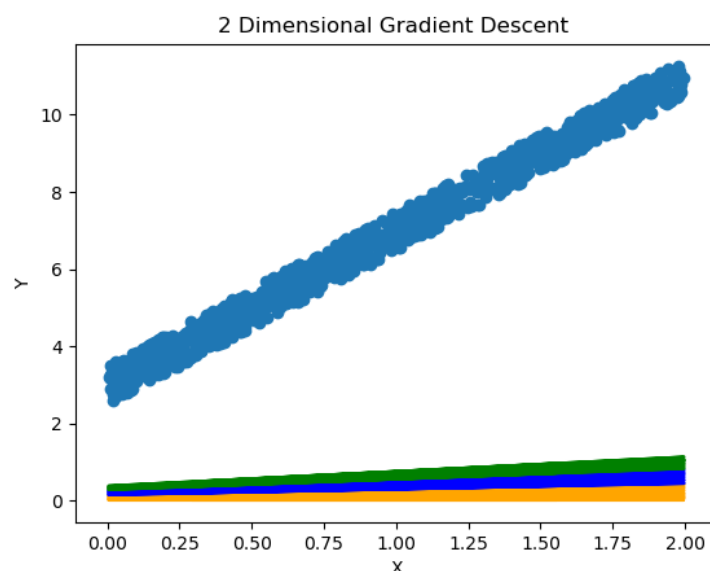
نکته ایی درباره نرخ یادگیری : تاثیر میزان نرخ یادگیری روی فاکتورهای نظیر سرعت همگرایی بسیار تاثیر گذار است به گونه ایی که در ادامه مشاهده خواهید نمود اگر میزان نرخ یادگیری را زیاد قرار دهیم (بیشتر از میزان Optimum ممکن است اصلا همگرایی رخ ندهد)

نرخ یادگیری بهینه یا به اصطلاح Optimum زمانی است که خط ما با یک نرخ نسبتا زیاد در ابتدا (فواصل دورتر از بهینه محلی یا خطوط نارنجی) به سمت نقطه ی بهینه محلی حرکت می کند و زمانی که به نقطه مینیمم محلی یا کمترین خطا نزدیک می شود حرکت آن کندتر میشود (خطوط ابی یا در انتها سبز)

در نتیجه همانطور که در فایل توضیحات کد توضیح داده شده است در ابتدا با میزان یادگیری 0.001 شروع می کنیم و توضیحات را می دهیم و در نهایت به میزان نرخ یادگیری 0.9 خواهیم پرداخت .

نرخ یادگیری = 0.001 :

همانطور که در شکل پایین مشاهده می کنید این میزان نرخ یادگیری بسیار کمتر از میزان بهینه بوده بطوری که حتی اگر میزان اپوک را از 30 به 100 نیز برسانیم حتی ممکن است همگرایی رخ ندهد :

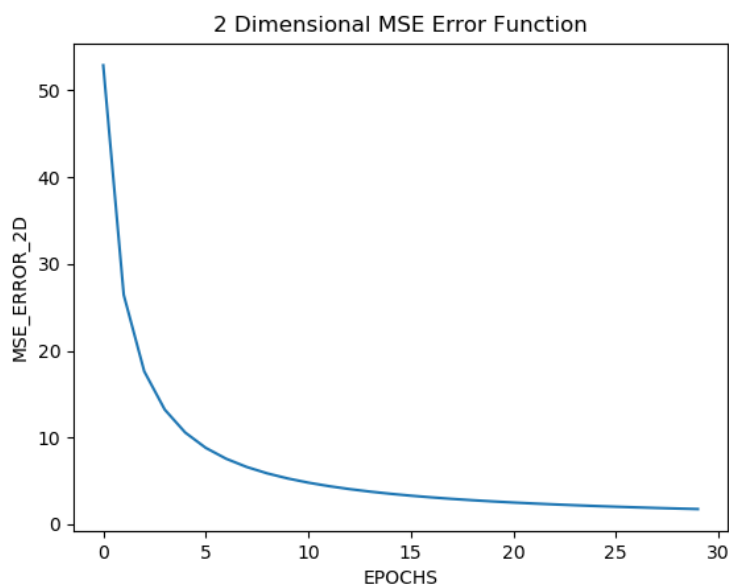


همانطور که در شکل بالا قابل مشاهده است تغییرات به ازای هربار تکرار را داریم ولی انقدر این تغییرات اهسته صورت می گیرد که نتوانسته به نقطه ی مینیمم محلی همگرا شود

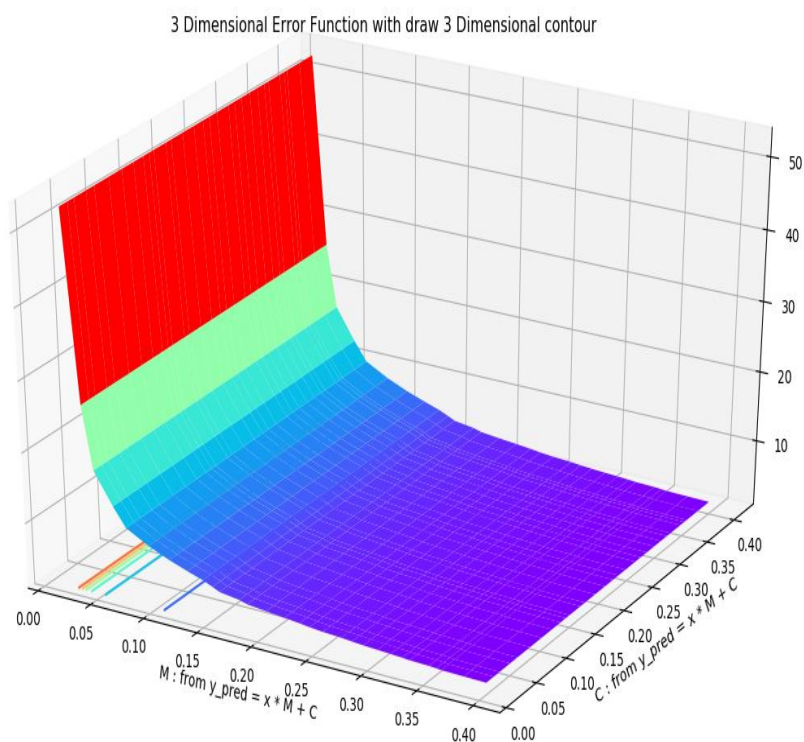
به طور کلی میتوان با توجه به صورت سوال ، از سه نوع تابع خطای متفاوت استفاده کرد که در ادامه به بررسی هر کدام از آنها می پردازیم :

تابع خطای MSE :

در نمای ۲ بعدی این تابع خطا به ازای نرخ یادگیری 0.001 کاملاً قابل مشاهده است که در هر بار آپدیت خط (هر اپوک) میزان خطا کاسته شده ولی باز هم با مینیمم مدنظر ما فاصله ی زیادی دارد :



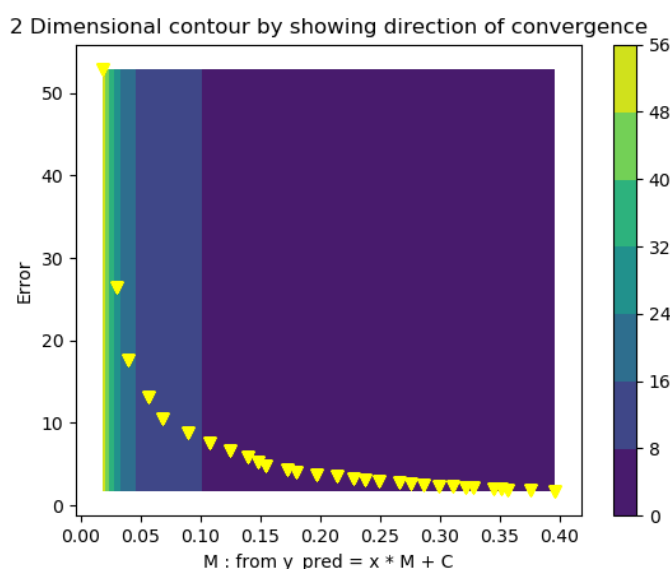
اگر این تابع خطا را به صورت ۳ بعدی نیز رسم کنیم خواهیم دید به ازای میزان تغییرات ضرایب معادله (m و c) به نقطه ی مینیمم محلی همگرا میشویم ولی بازهم این مقدار از مقدار مورد انتظار ما بیشتر است :



خطوط کانتور سه بعدی در زیر نمودار ۳ بعدی قابل مشاهده است ، همانطور که در شکل بالا مشاهده می کنید تغییرات در هر مرحله (میزان فواصل خطوط رنگی و طول آنها) بسیار کم بوده و با سرعت بسیار کمی به سمت مینیمم محلی در حال همگرا شدن میباشد .

همانطور که از خط قرمز رنگ در نمودار ۳ بعدی برداشت میشود در گام اول با سرعت بیشتر به سمت مینیمم محلی حرکت کرده است ولی در گامهای بعدی این سرعت به کندی انجام گرفته است به طوری که به اون نقطه ی مینیمم محلی اصلا همگرا نشده است (طول رنگ قرمز بسیار بزرگ تر از سایر رنگها میباشد در نتیجه سرعت ان بیشتر بوده است)

اگر نمودار کانتور دو بعدی ان را نیز ترسیم کنیم ، فلش های زرد رنگ مسیر حرکت به سمت مینیمم محلی را به خوبی به ما نشان میدهد :

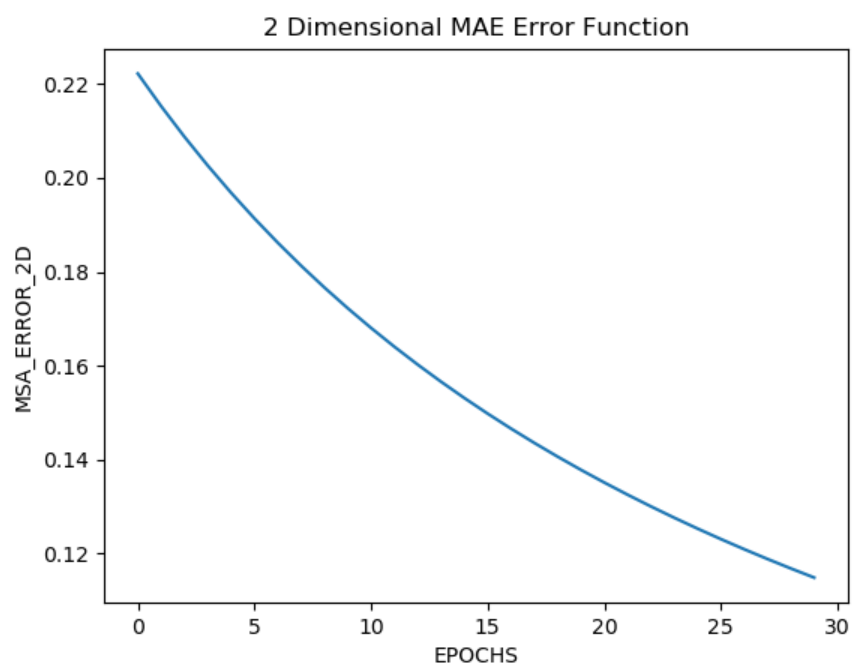


همانطور که در شکل بالا قابل مشاهده است فاصله ی اولین فلش زرد رنگ با دومین فلش زرد زنگ نیز بیشتر از سایر فلش ها است که همین صحت حرفهای ما را در شکل قبلی تایید می کند .

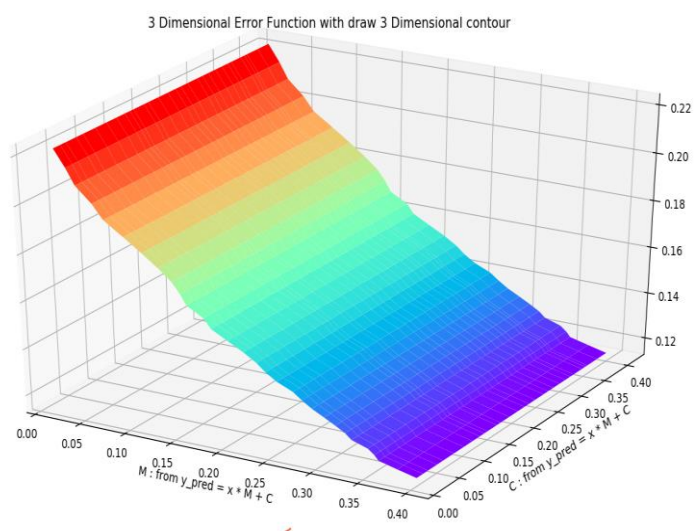
تابع خطا MAE :

تنها تفاوت این تابع خطا با تابع خطای قبلی در این است که در تابع خطای قبلی اختلاف میزان γ پیش بینی شده از γ واقعی کم می شود و آن را به توان ۲ می رساند (دلیل وجود توان ۲ به خاطر این است که اگر گاهی میزان γ پیش بینی شده از میزان γ واقعی کمتر باشد مقدار خطا منفی نشده که در ادامه باعث رخ دادن خنثی شدن میزان خطاهای مثبت و منفی با یکدیگر نشود) در این تابع از قدرمطلق به جای توان ۲ استفاده شده است.

علاوه بر تفاوت گفته شده که به ظاهر در استفاده از علائم ریاضی بود ، تفاوت اصلی و مهم دیگر در تقابل با داده های نویز میباشد . اگر داده های نویز داشته باشیم تابع خطای MAE میزان خطای درستی را نشان نمیدهد و به اصطلاح رفتار درستی را از خود نشان نمیدهد و باعث میشود گاهی به اون بهترین خط ممکن نرسیم درحالی که MSE به هر صورت به اون بهترین خط همگرا خواهد شد که البته نباید از میزان تاثیر گذاری نرخ یادگیری غافل شد .

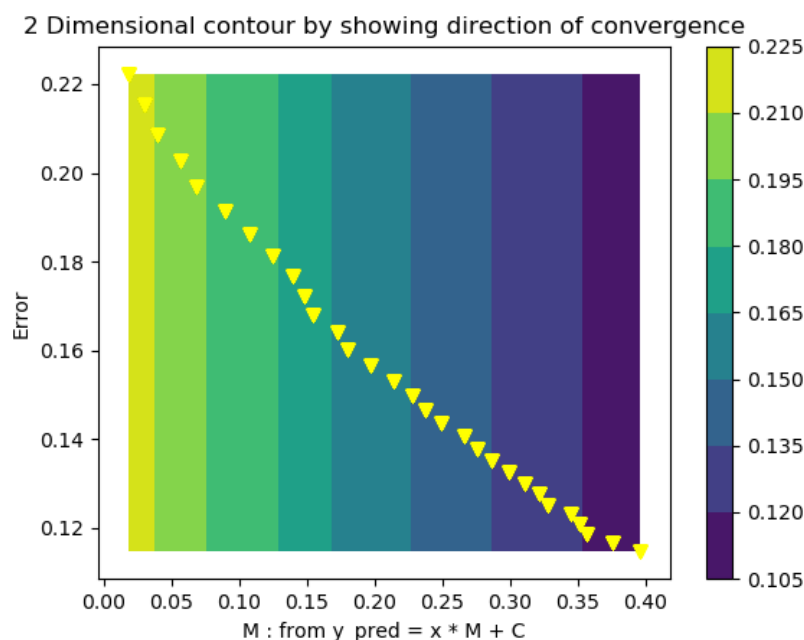


نمودار خطای ۳ بعدی این تابع خطا نیز به صورت زیر است :



تفاوت بعدی که در تصاویر سه بعدی قابل مشاهده است ، به دلیل وجود توان ۲ در تابع مربوط به MSE این تابع به شکل تابع درجه ی دوم رسم میشود در حالی که در تابع خطای MAE این شکل به صورت خط رسم می گردد .

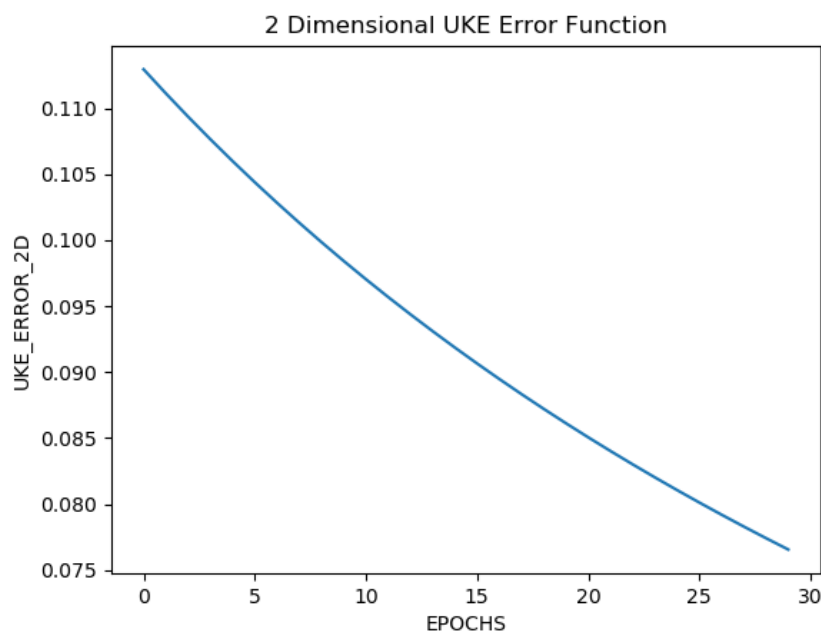
شکل ۲ بعدی کانتور این تابع نیز به ازای همین نرخ یادگیری به شرح زیر است :



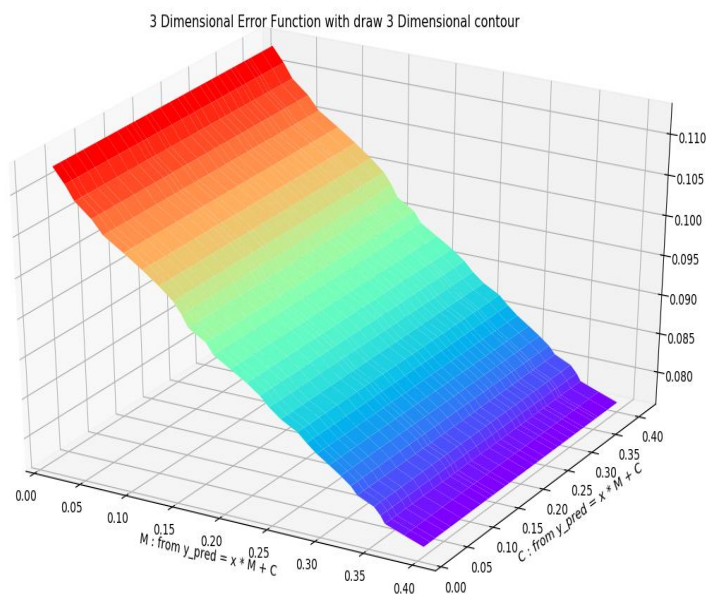
همانطور که در این شکل قابل مشهود است صحت صحبت های قبلی گفته شده را تایید می کند

خطای UAE :

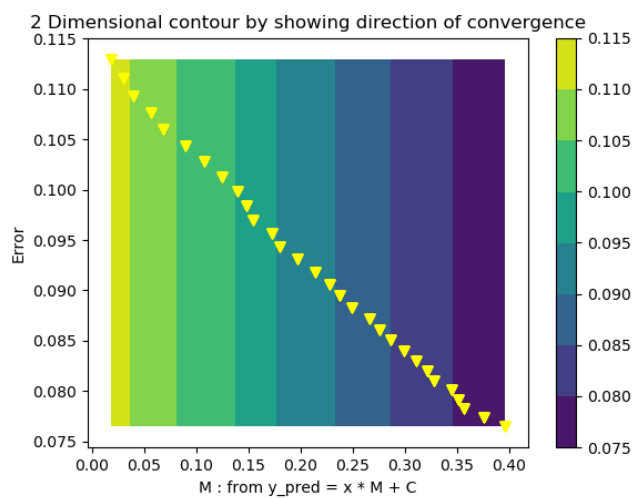
تفاوت این معیار خطا با معیارهای قبلی در این است که در اینجا به منفی یا مثبت شدن میزان خطا توجهی نمی شود که همین امر باعث میشود زمانی که نرخ یادگیری ما افزایش می یابد این خطا حتی منفی نیز شود . نکته ی دیگر در رابطه با این تابع خطا این است که به هیچ وجه قابل استفاده برای نمایش همگرایی به سمت بهینه محلی مفید نیست زیرا اثرات خطا ها را به دلیل رخ دادن منفی و مثبت با یکدیگر خنثی می کند



نمودار ۳ بعدی ان نیز به شرح زیر است :



نمودار کانتور ۲ بعدی ان نیز به صورت زیر است :



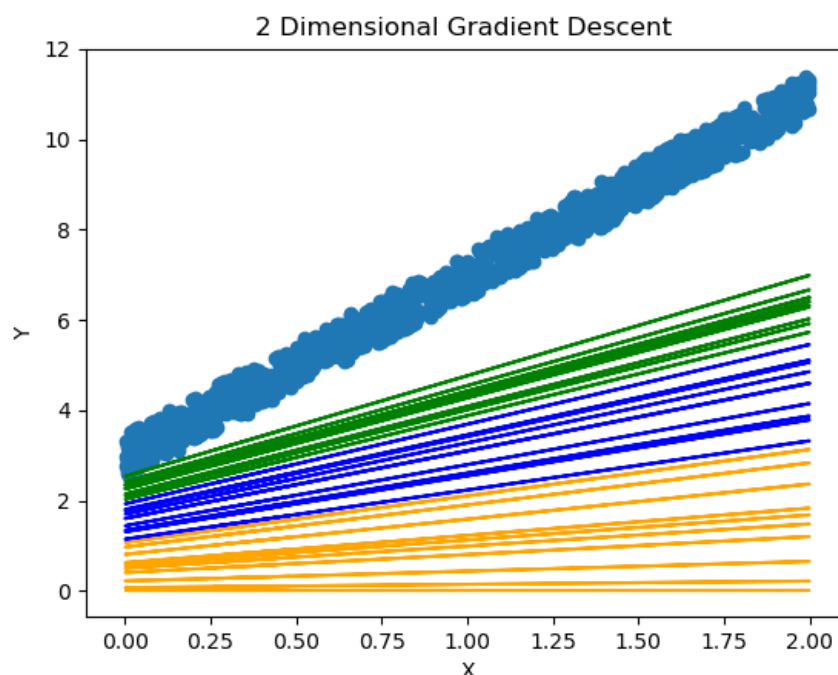
و در آخر نیز میزان ضرایب بدست آمده به ازای آخرین اپدیت به شرح زیر است:

```
Optimum m in "y = m * x + c" is [0.39606817]  
Optimum c in "y = m * x + c" is [0.40857448]
```

نرخ یادگیری = ۰/۰۱ :

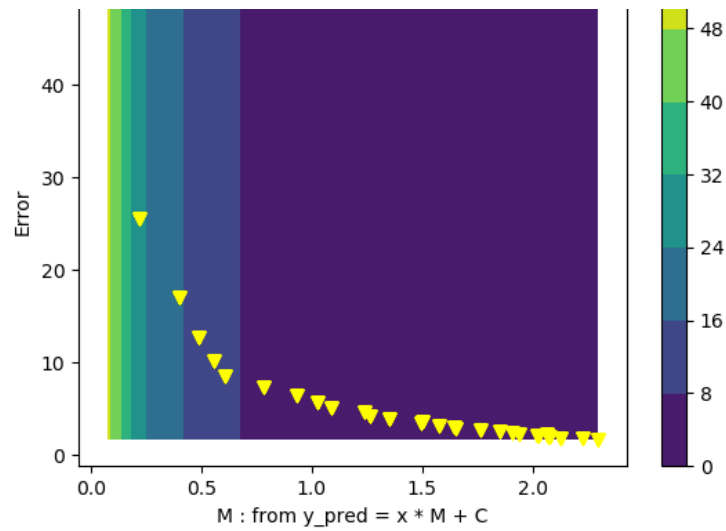
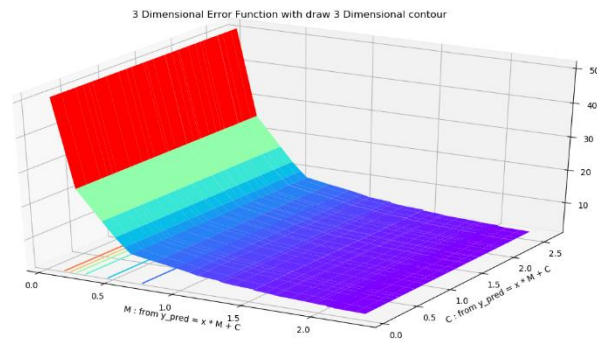
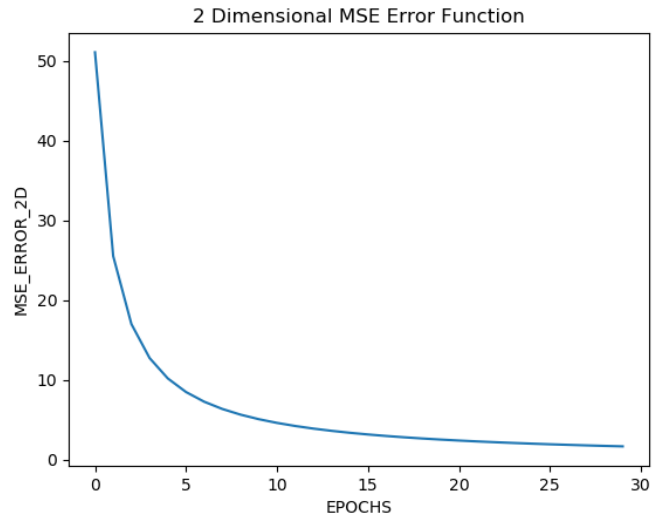
این میزان نرخ یادگیری برای این شکل به عنوان Optimum در نظر گرفته می شود بدین صورت که همانطور که در ادامه مشاهده خواهیم کرد از این میزان نرخ یادگیری بیشتر موجب عدم همگرایی یا نوسانات شدید در کنار نقطه ی مینیمم محلی می شود و از این مقدار کمتر نیز انقدر سرعت حرکت کند خواهد بود که ممکن است هرگز به نقطه ی بهینه محلی همگرا نشویم یا در تعداد اپوک های زیادی به همگرایی برسیم

همانطور که در شکل پایین قابل مشاهده است تغییرات در ابتدا زیاد و در
اپوک های آخری تغییرات با سرعت کمتری صورت می گیرد و در انتها به
کمترین میزان خطا همگرا می شود :

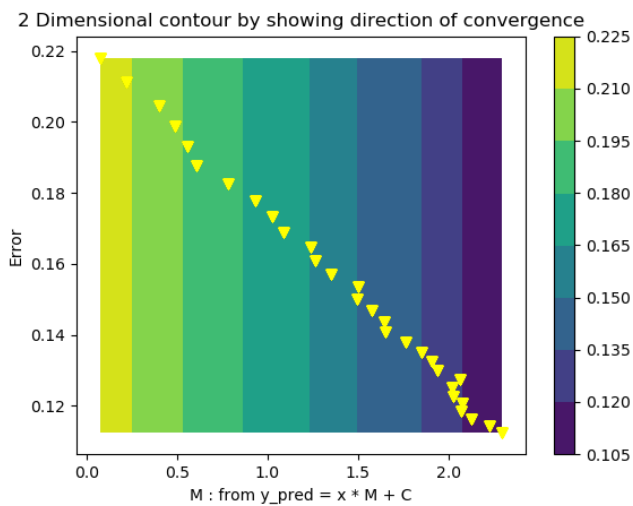
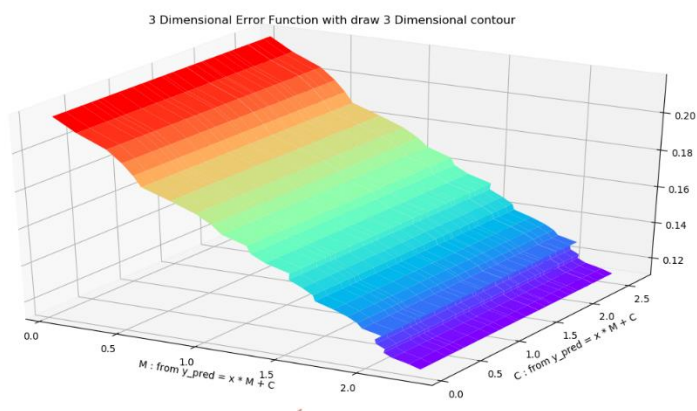
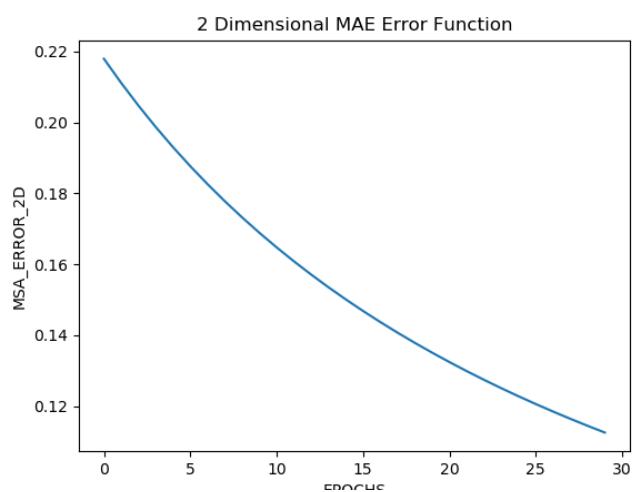


بقیه ی توضیحات گفته شده در قسمت های قبلی در این قسمت نیز یکسان
است پس از تکرار نکات گفته شده خودداری می کنیم و تنها شکل های
ترسیم شده را مشاهده می کنیم

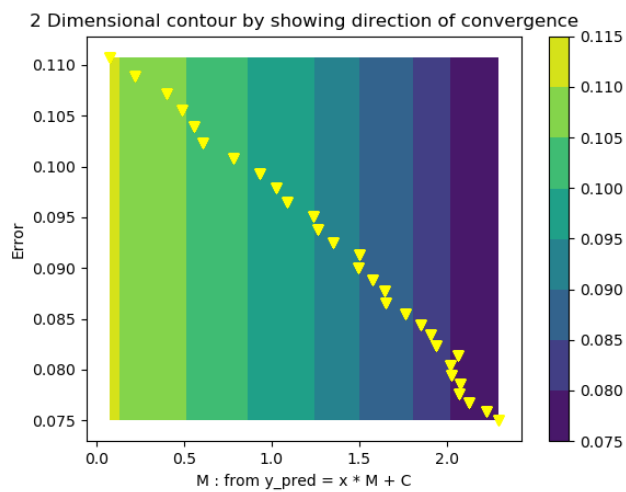
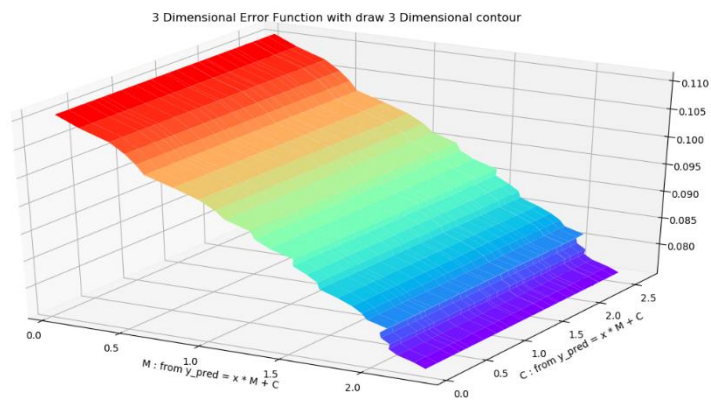
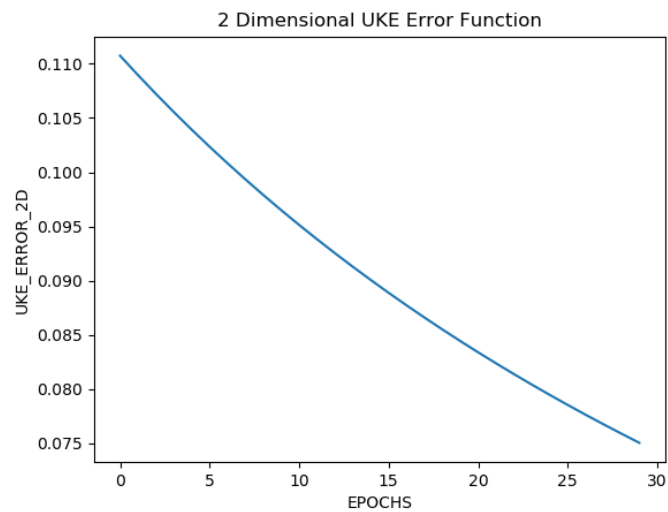
تابع خطا MSE :



تابع خطا MAE :

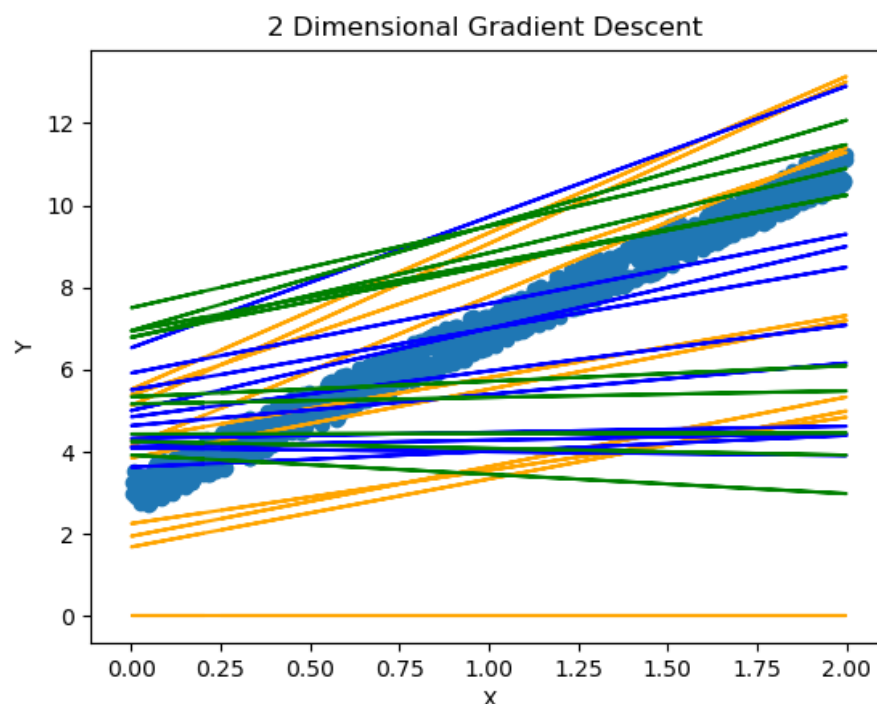


تابع خطا UKF :



حال به بررسی نرخ های یادگیری بیشتر از Optimum می پردازیم ، در این صورت انقدر نوسانات تغییرات هر خط به ازای هر اپوک زیاد خواهد بود که به ازای نرخ یادگیری بالاتر از 0.2 خواهیم دید که مینیمم خطا نخواهیم رسید .

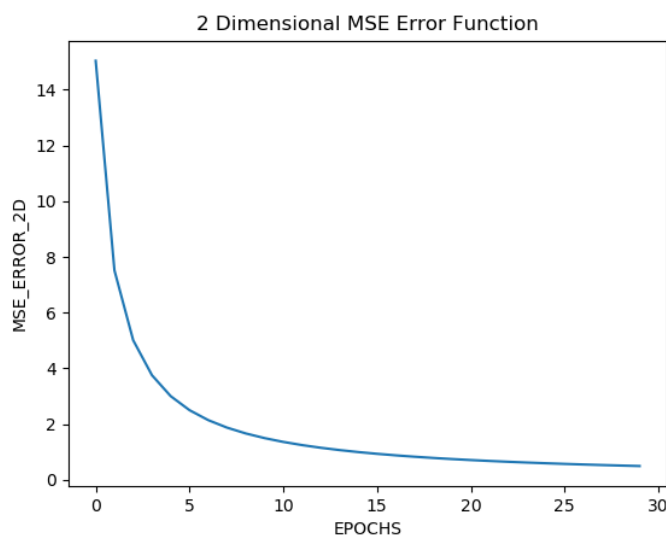
نرخ یادگیری = 0.2 :



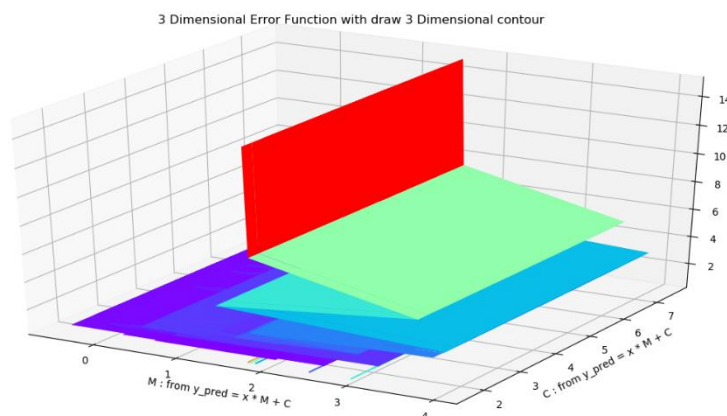
همانطور که در شکل بالا قابل مشاهده است ، به جای اینکه با شیب ملایم به سمت خط بهینه نهایی همگرا شویم ، تغییرات بسیار زیاد خواهد بود و در انتها نیز همگرایی رخ نمی دهد .

در ادامه به بررسی توابع خطا در صورت قرار دادن نرخ یادگیری بیشتر از حالت Optimum می پردازیم

تابع خطا MSE :

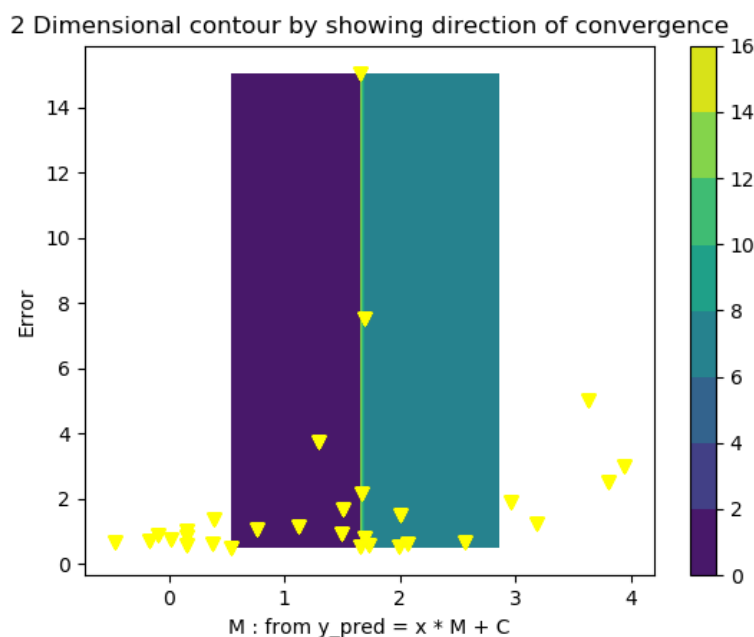


به دلیل اینکه در ابتدا خط شروع با سرعت بسیار زیادی به سمت مینیمم محلی حرکت میکند در نتیجه در شروع کار به ظاهر با خطای کمتری شروع می کنیم ولی سپس با ادامه ی این روند ، و نوسانات بیش از حد به اون مینیمم محلی نخواهیم رسید ولی با این حال میزان MSE کاهش میابد



در شکل سه بعدی تابع خطا میزان تغییرات در هر بار اپدیت بسیار بهتر از شکل دو بعدی دیده خواهد شد ، نکته ایی که به وضوح قابل مشاهده است این است که در ابتدا (قسمت قرمز) به صورت نزولی مقدار خطای ان کاهش یافته است و سپس به صورت نوسان های شدید این تغییرات کاهش داشته است به طوری که اگر نقطه ی اخر که ان را اخرین اپدیت m و c است را نقطه ی بهینه ی محلی بنامیم (با نرخ یادگیری 0.2) قابل مشاهده است که اطراف نقطه ی بهینه محلی نوسان ها با شدت بسیار زیادی نسبت به اوایل این تغییرات بوده است .

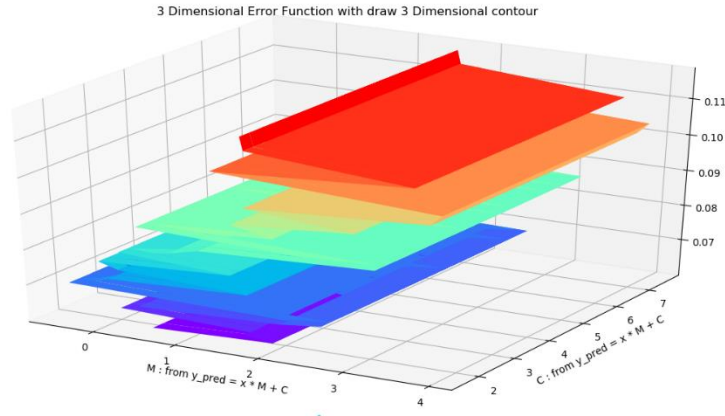
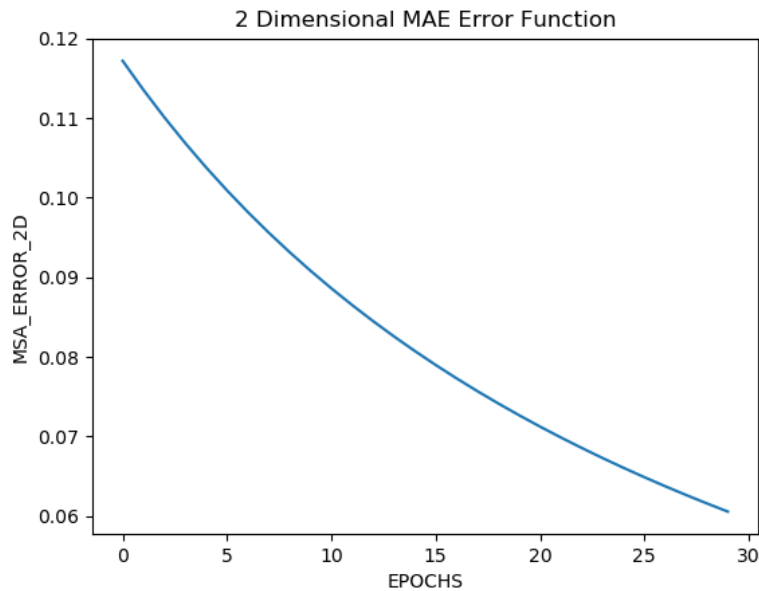
بر روی کانتور ۲ بعدی زیر نیز میتوان این نوسانات اطراف بهینه محلی را مشاهده نمود :



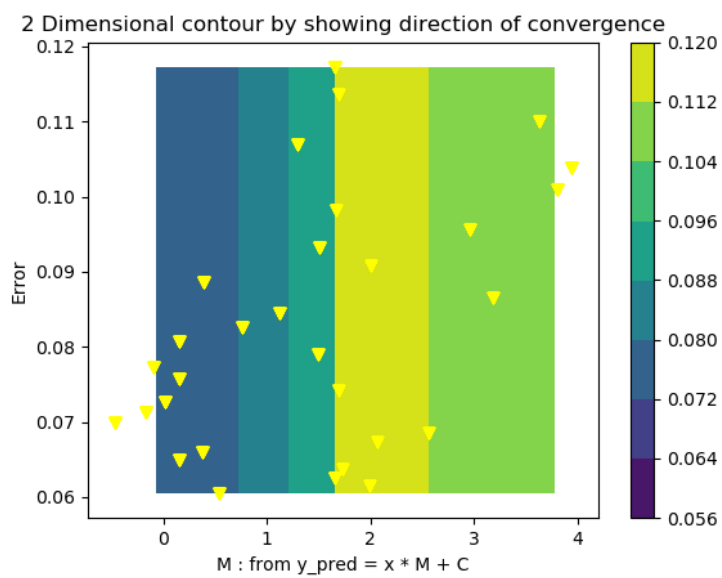
همانطور که به وضوح بیشتر نسبت به دو شکل قبلی در این شکل قابل مشاهده است ، فلش های زرد رنگ در اطراف بهینه ی محلی نوسانات بسیار شدیدی دارند و این بخاطر بیشتر بودن نرخ یادگیری نسبت به مقدار Optimum میباشد .

تحلیل توابع MAE و UKL نیز مثل MSE است که در ادامه به طور خلاصه تر به آن میپردازیم

تابع خطای MAE :

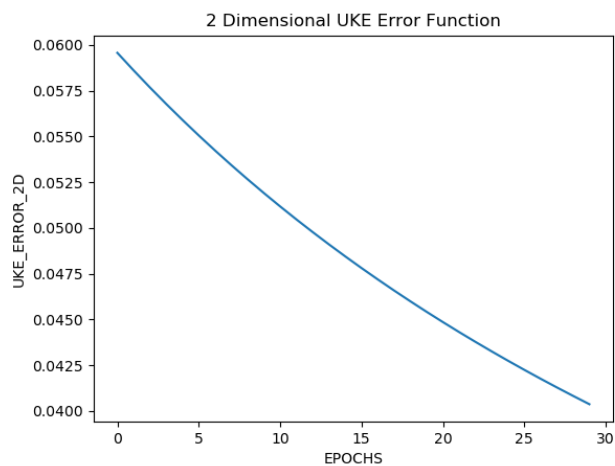


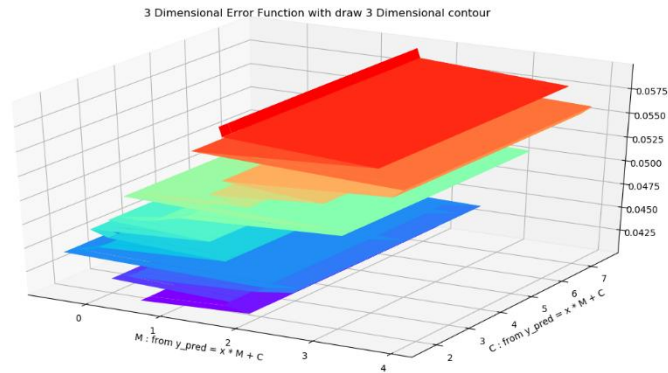
تفاوتی که بیشتر در شکل سه بعدی دیده می شود این است که شدت نزولی بودن خطا که در قسمت قرمز رنگ در قسمت قبل قابل مشاهده بود در این قسمت قابل مشاهده نیست و نوسانات از همان ابتدا به صورت شدید تری نسبت به تابع خطای قبلی رخ داده است .



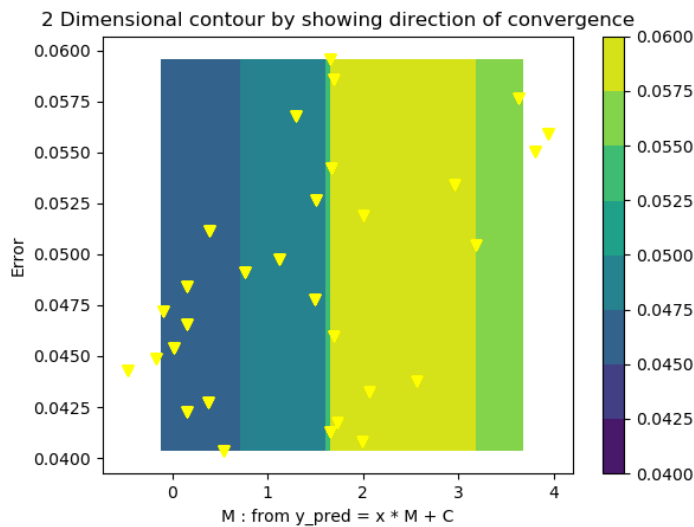
در کانتور فوق نیز صحت حرفهای زده شده قابل مشاهده است .

تابع خطا UKE :



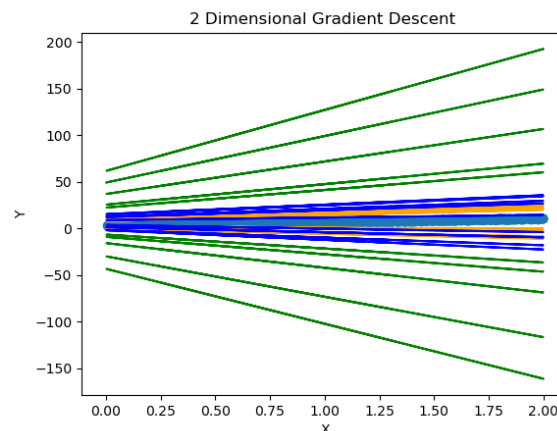


در ظاهر دو شکل مربوط به MAE و UKE باهم تفاوتی ندارند ولی این تنها با این میزان نرخ یادگیری رخ داده است ولی برای نرخ های یادگیری بالاتر خواهیم دید که تفاوت این ۲ تابع خطا نیز به وضوح قابل مشاهده است .

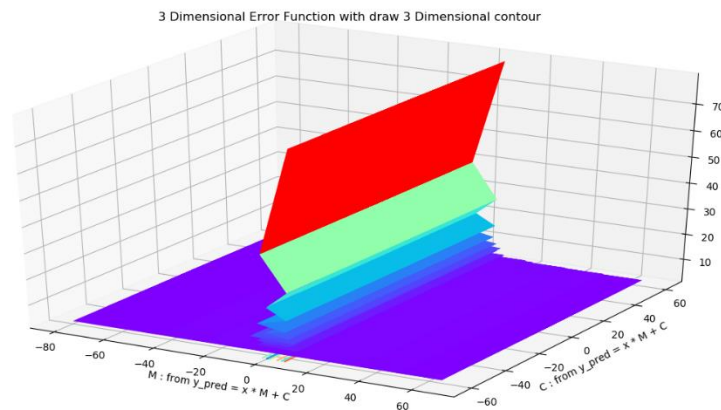
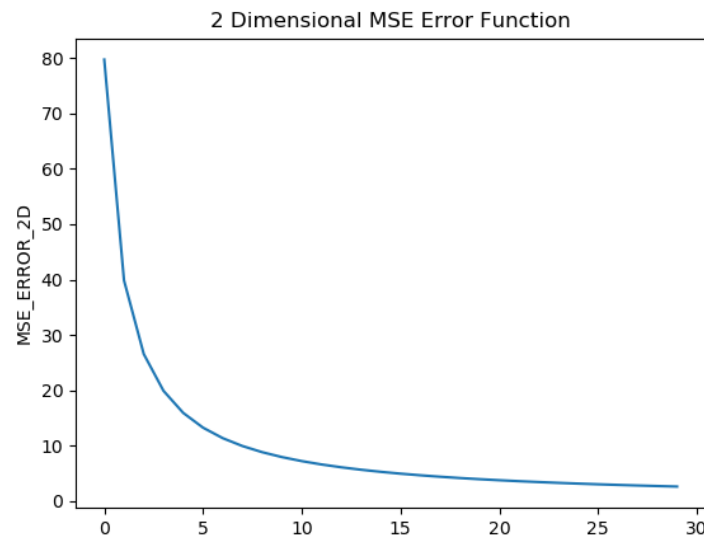


نرخ یادگیری = ۰/۵ :

حال اگر نرخ یادگیری را بسیار بزرگتر از مقدار Optimum قرار دهیم نوسانات بسیار شدید تر رخ خواهد داد به طوری که شکل گرادیان محلی ان به صورت زیر خواهد شد . در این شکل قابل مشاهده است زمانی که نرخ یادگیری را در این حالت قرار دهیم در اپوکهای های اول در اثر نوسانات شدید به نقطه ی مینیمم همگرا میشویم (خط های نارنجی و ابی) سپس به این دلیل که میزان نوسانات زیاد است دوباره از نقطه ی بهینه محلی فاصله میگرد ، شما به عنوان مثال فرض کنید دره ای را در نظر داریم ، زمانی که با شدت بسیار زیاد توپ را به دیواره های ان بکوبیم ابتدا به سمت پایین حرکت خواهد کرد (نقطه بهینه) ولی وقتی به پایین دره نزدیک میشود در اثر کاهش فاصله ی دیوارهای دره باهم ، توپ باز هم با تعداد برخوردهای بسیار بیشتر نسبت به حالت اولیه به دیواره ها برخورد کرده و به سمت بالا (دور شدن از بهینه) حرکت می کند با این مثال شکل پایین کاملاً قابل درک خواهد بود

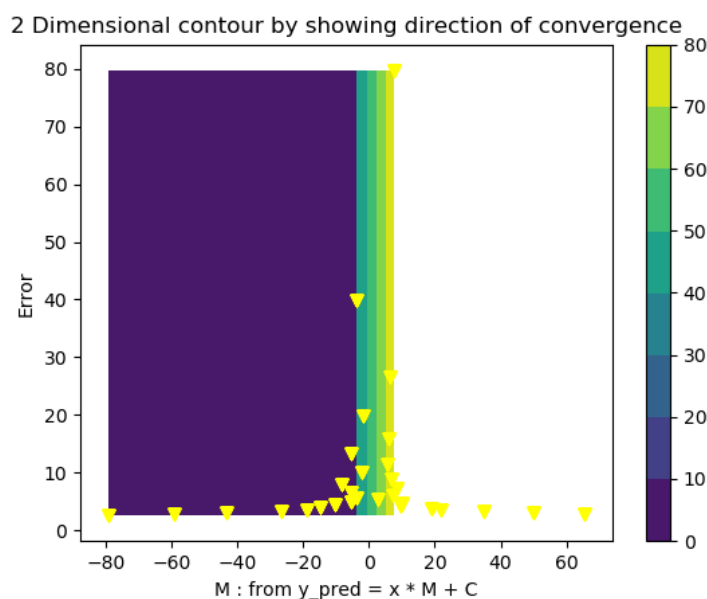


تابع خطا MSE :



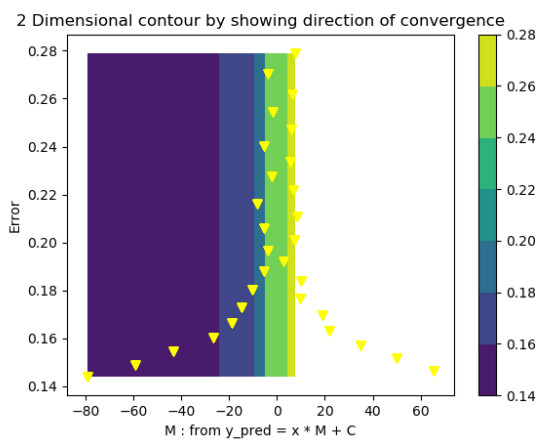
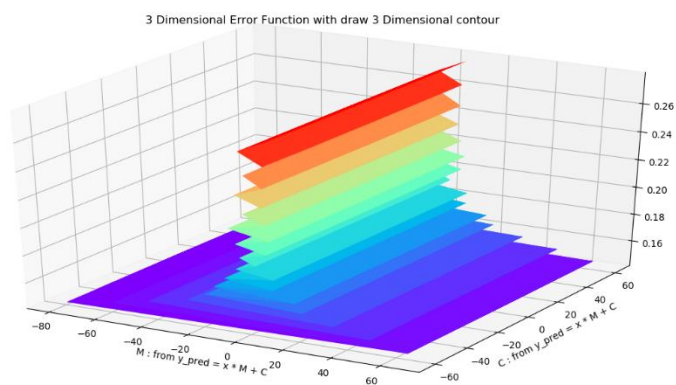
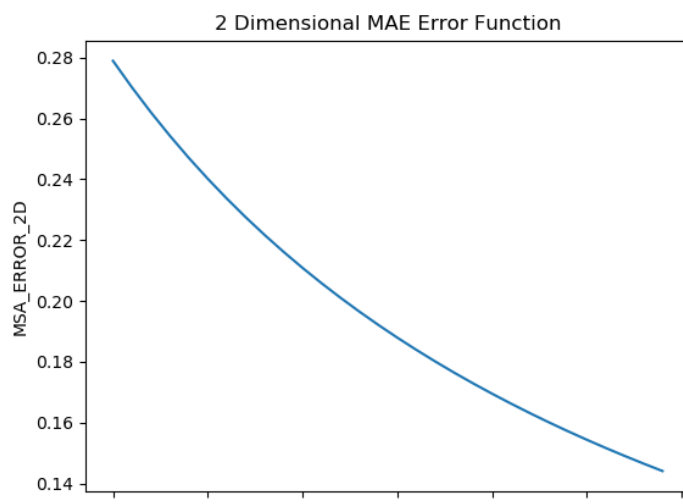
همانطور که در شکل سه بعدی تابع خطا بشدت قابل مشاهده است در ابتدا با سرعت بسیار زیاد به سمت بهینه ی محلی حرکت کرده ولی زمانی که به این نقطه نزدیک می شود به شدت میزان نوسانات زیاد بوده به طوری که کل صفحه ی منتهی به مینیمم محلی را با رنگ ابی پوشانده است ، اگر میزان

اپوک را افزایش میدادیم میتوانستیم ببینیم که دوباره با افزایش اپوک ها با
دور شدن خطوط دوباره میزان خطا به طور ثابلی ملاحظه ایی افزایش می یافت

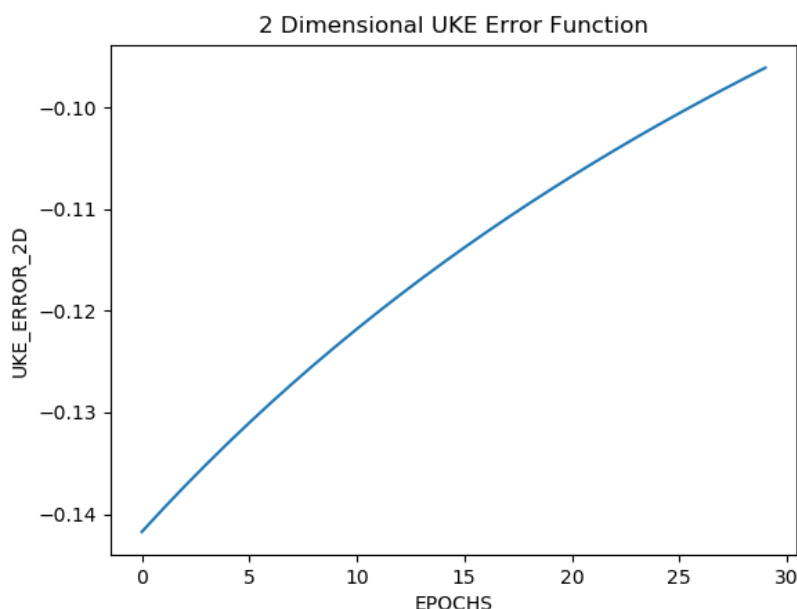


با ملاحظه ی تصویر دو بعدی کانتور فوق میتوان دریافت که در اطراف بهینه
محلی تغییرات بشدت زیاد خواهد بود طوری که به مینیمم محلی همگرا
نخواهیم شد . (هرچند میزان خطا کاهش یافته است !)

تابع MAE :



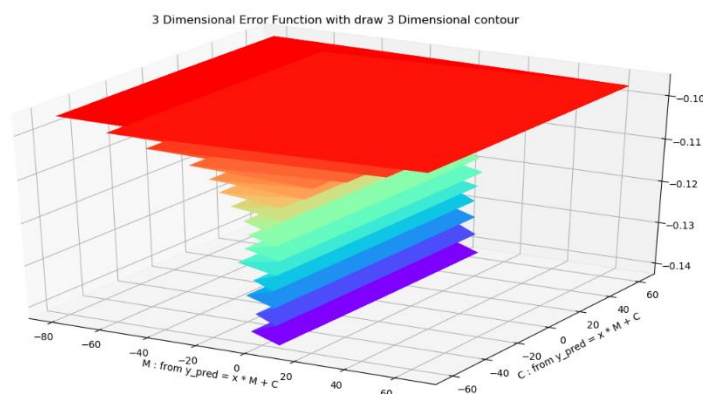
تابع UKE :



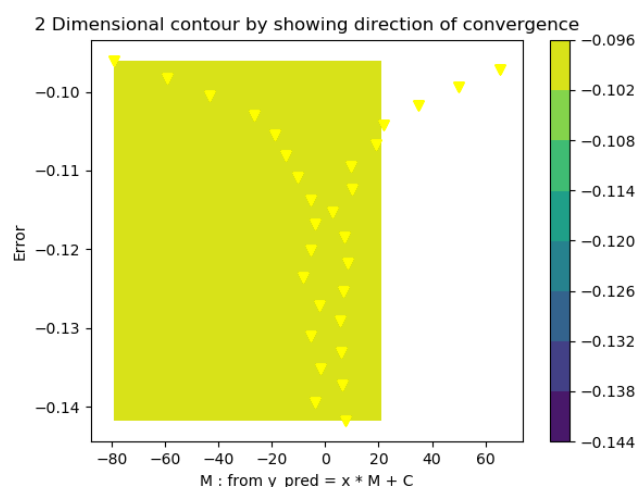
دلیل شکل فوق چیست؟! پاسخ این سوال را این گونه جواب خواهیم داد . در فرمول مربوط به این تابع خطا بدلیل عدم وجود توان ۲ یا قدرمطلق به راحتی میزان خطای مثبت و منفی با یکدیگر جمع شده و میتوانند در کل خطای منفی را تولید کنند اما چه زمانی پاسخ خطای ما منفی خواهد شد؟! زمانی که مقدار پیش بینی شده از مقدار واقعی طبق فرمول بیشتر شود این خطا منفی خواهد شد

بحث نمودیم در ابتدای کار ، زمانی که میزان نرخ یادگیری ما از حدی بیشتر شود تغییرات در هر اپوک بسیار زیاد خواهد بود به طوری که در شکل مربوط به گرادیان ملاحظه کردیم در ابتدا خط ایدیت شده بالاتر از حتی داده ها قرار خواهد گرفت لذا میزان پیش بینی شده از میزان واقعی بیشتر شده در نتیجه

خطای منفی را تولید خواهد کرد. ولی به مرور زمان و گذشت هر اپوک کم کم به خطای صفر نزدیک خواهیم شد

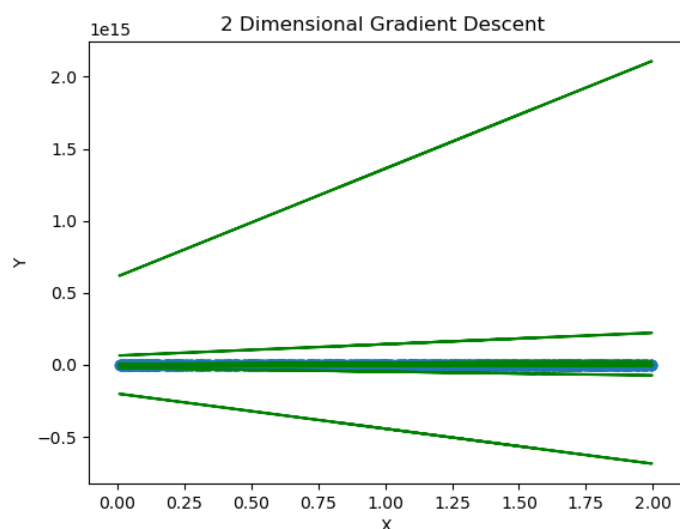


همانطور که از شکل ۳ بعدی تابع خطا نیز قابل مشاهده است تغییرات در ابتدا بسیار شدید رخ خواهد داد و سپس هر چه به مینیمم محلی نزدیک می شویم این تغییرات آهسته تر رخ خواهد داد ، این شکل به خوبی صحبت زده شده را نشان می دهد



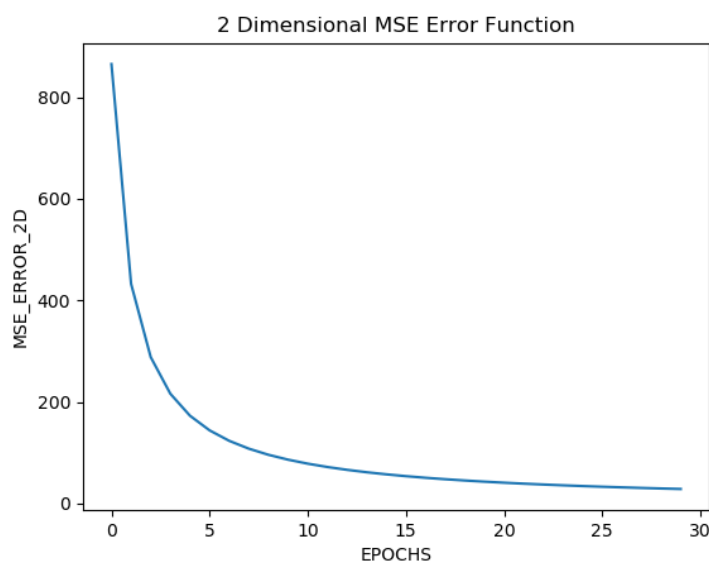
نرخ یادگیری = ۰/۹ :

باتوجه به شکل زیر میتوان دریافت که با قرار دادن این نرخ یادگیری شکل گرادیان محلی ما بر اثر نوسانات بیش از حد به یک خط مپ می شود :

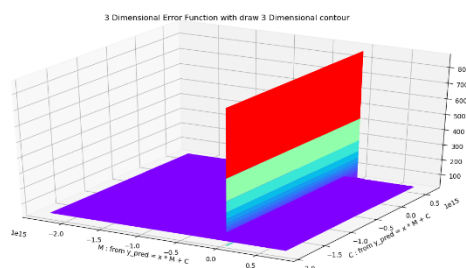


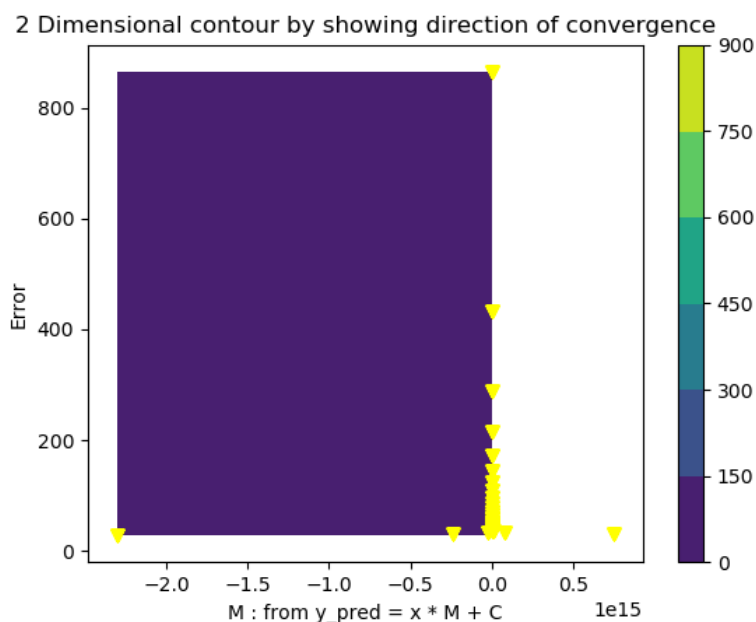
تنها تفاوت این قسمت با قسمت قبلی در این است که میزان نوسانات تغییرات اطراف گرادیان محلی بسیار بیش از حد اندازه است که در پایین شکل های مرتبط به توابع خطا و کانتورهای آنها را مشاهده می کنیم

تابع MSE :



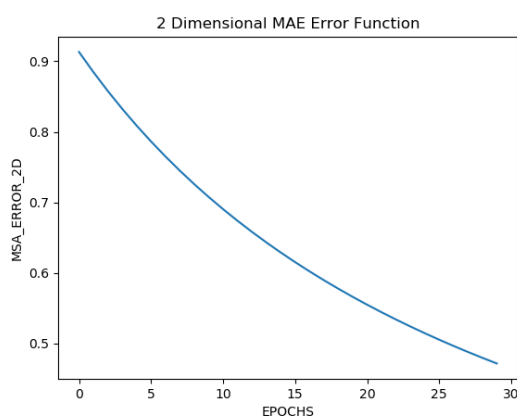
با بررسی دقیق تر اشکال مربوط به توابع MSE و MAE میتوان دریافت زمانی که میزان نرخ یادگیری را بیش از حد Optimum قرار می دهیم ، در ابتدای کار با میزان خطای بسیار بیشتر از میزان خطا با نرخ یادگیری Optimum شروع کرده ولی در نهایت به مینیمم خواهیم رسید ، هرچند که اگر تعداد اپوک ها را افزایش دهیم خواهیم دید که دوباره خطا افزایش پیدا خواهد کرد .

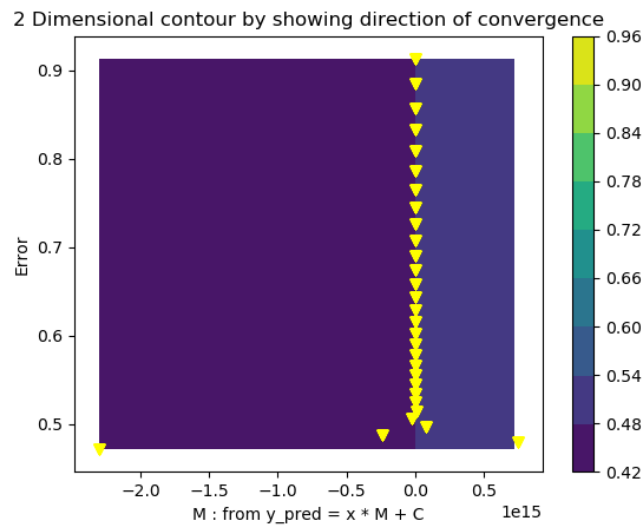
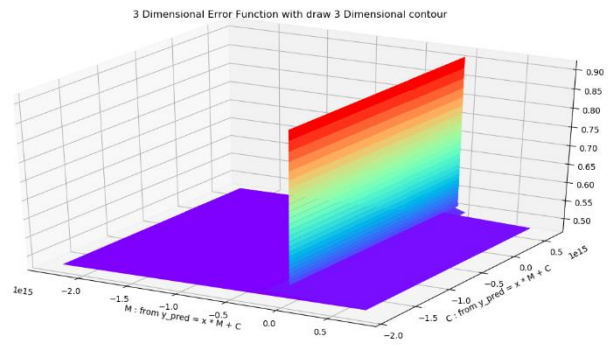




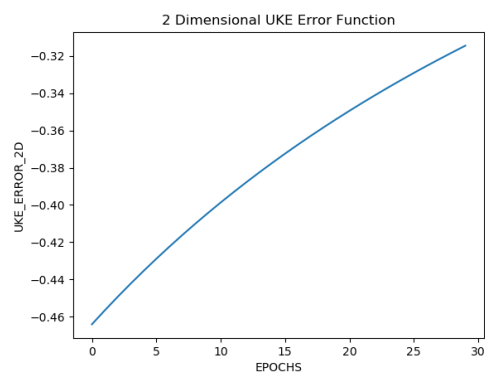
همانطور که در شکل بالا مشاهده می کنید در ابتدا به خاطر نرخ یادگیری بسیار زیاد به صورت نزولی و با سرعت زیاد به سمت بهینه سراسری حرکت میکنیم و در نقاط نزدیک بهینه محلی یا سراسری ، بشدت نوسانات زیاد است که روی شکل نیز قابل مشاهده است .

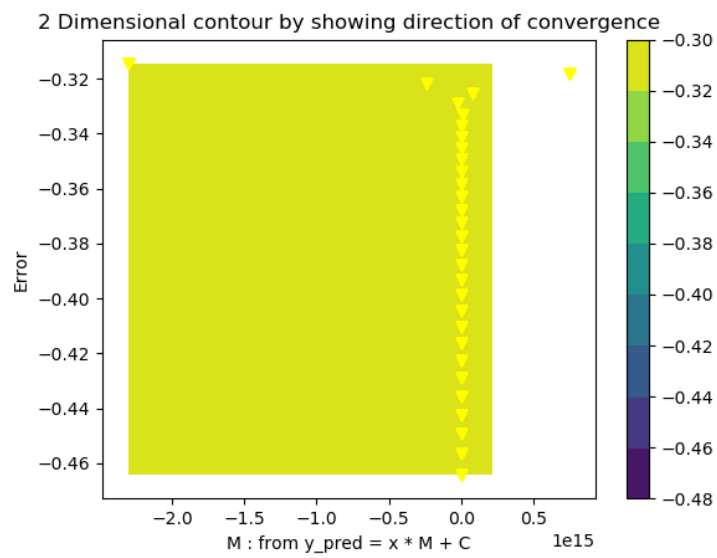
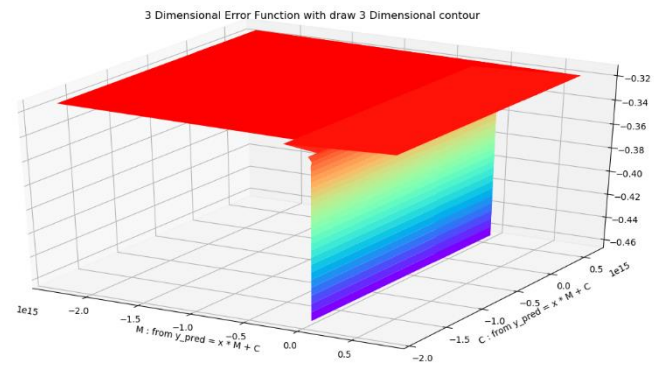
تابع MAE :





تابع UKE :





جمع بندی :

تمامی نکات ذکر شده در تمامی ۱۶ دیتاست بررسی شده یکسان است

با این تفاوت که در بعضی دیتاست ها نقطه Optimum برای نرخ یادگیری ۰/۰۱ و در برخی دیگر میزان ۰/۰۰۱ است ولی تمام تفاسیر گفته شده یکسان است

به این ترتیب که اگر میزان نرخ یادگیری کمتر از میزان Optimum باشد همگرایی با سرعت بسیار پایین صورت می گیرد (در تعداد اپوک های زیاد) یا اصلا صورت نمیگیرد و اگر میزان نرخ یادگیری را بیشتر از میزان Optimum محلی قرار دهیم ، نوسانات افزایش خواهد یافت و هرچه این میزان را افزایش دهیم تا به ۰/۹ برسیم تغییرات بشدت قابل ملاحظه است به طوری که اشکال مربوط به گرادیان کاهشی ما به یک خط مپ می شود

تمامی اشکال مربوط به دیتاست ها و نتایج بدست آورده شده در فایل زیپ به همان صورت گفته شده قرار داده شده است .

در پایان از زحمات تمامی دوستان تشکر و قدردانی می نمایم

پایان