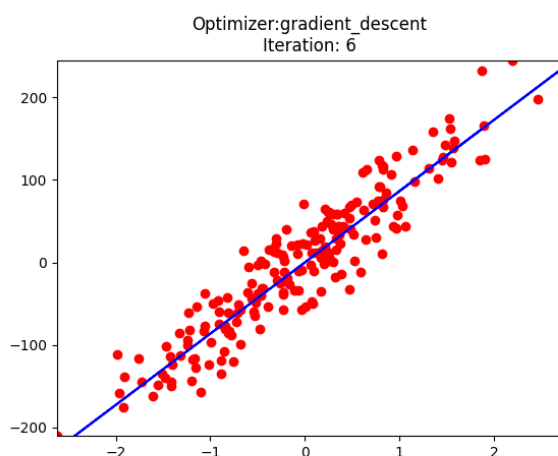


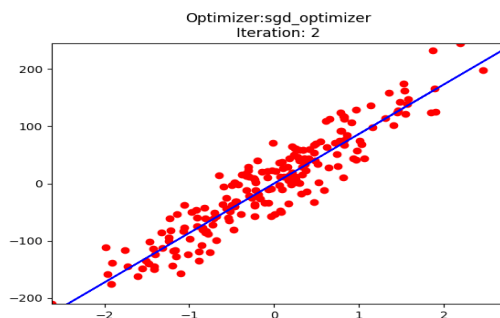
در این تمرین با پیاده سازی اپتیمایزر های مختلف روی دیتاستی که وجود دارد مواجه هستیم. هر کدام از اپتیمایز ها را با هایپر پارامترهایی که با آنها بهترین نتیجه را داشته اند گزارش می کنیم و آنها را با بقیه مقایسه خواهیم کرد.

قبل از اینکه سراغ پیاده سازی برویم نکته ای وجود دارد که آن است که برای هر مدل تا زمانی که خطا کاهشی است مدل را ادامه دادیم تا بیش از حد هزینه محاسباتی بالایی نداشته باشیم. حالا نوبت آن است که از اولین اپتیمایزر شروع به بررسی می کنیم.

مستقل از توضیح کد برای اپتیمایز gradient descent بهترین مقدار برای هایپر پارامترش که α یا همان لرنینگ ریتمان بود میزان 0.003 بود که برخلاف میزان های بیشتر که کاملاً اشتباه تشخیص می دادند و همچنین میزان های کمتر از این مقدار که تا رسیدن به نقطه بهینه زمان طولانی صرف می کردند در 6 گام به میزان $loss = 933.0112$ رسید که میزان قابل قبول و خوبی به نظر می آید. نمودار زیر نمایانگر خط فیت شده روی داده هایمان است.

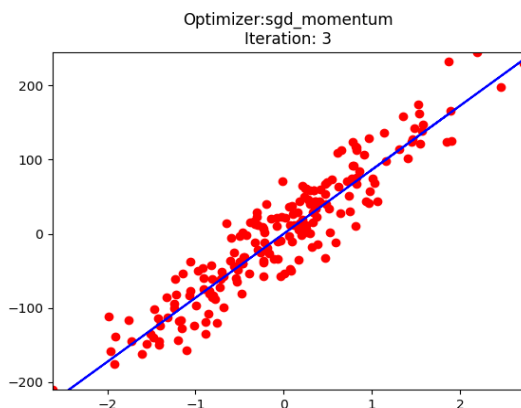


اپتیمایزر بعدی stochastic gradient descent یا به اختصار sgd است که نسبت به gd با batch کردن باعث می شود زودتر به مقدار بهینه برسیم که مطمئناً سائز batch نیز بسیار مهم است. با لرنینگ ریتم 0.01 و $batch-size = 20$ در 2 گام به مقدار $loss = 933.0177$ رسیده ایم که بسیار مناسب است. نمودار مرحله آخر را در زیر می بینیم.



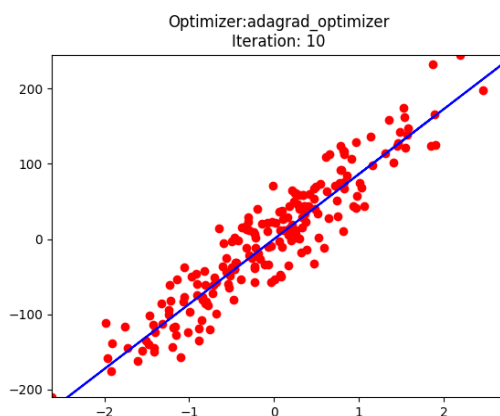
روش بعدی که sgd با پارامتر momentum است کمک می کند تا سریعتر به میزان بهینه برسیم و سرعت روند لرنینگ را افزایش می دهد. با هاپیر پارامترهای لرنینگ ریت 0.0009 و مومنتم 0.9 پس از 3 گام به میزان $loss = 933.0177$ رسیدیم.

نکته ای که نسبت به sgd عادی دارد آن است که به نظرم با لرنینگ ریت بسیار پایینتر توانستیم به مقدار خطای بهتری برسیم. این مورد در دیتاست های سخت تر بیشتر به چشم می آید زیرا باید پس از چندین بار ازوم و خطا الگوی لازم برای کلاس بندی یا رگرسیون را پیدا کنیم و این روش بهتر خواهد بود. نمودار زیر نمایانگر آخرین گام این روش می باشد.

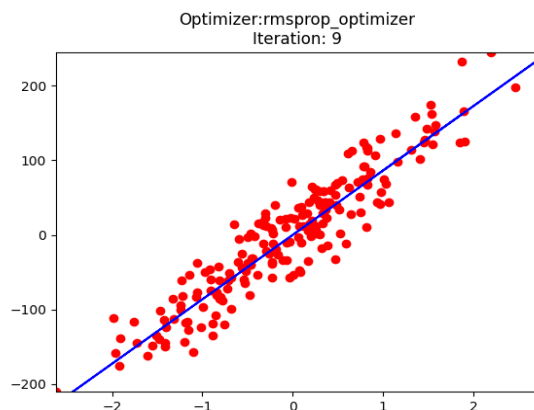


روش بعدی که adagrad است روند یادگیری بسیار کندی دارد و هر چقدر هاپیر پارامترها را تغییر دادم به میزان دلخواهی برای جواب بهینه نرسیدم. از آنجایی که با این اپتیمایزر آشنایی زیادی نداشتم با تحقیق راجب مقدار هاپیر پارامترهایش به نتیجه مناسبی نرسیدم اما از آنجا که سرعت لرنینگ بسیار پایینی داشت زمانی که لرنینگ ریت را 135 و اپسیلون را 0.01 در نظر گرفتم و در 10 گام به مقدار $loss = 933.0111$ رسیدیم.

به نظرم این روش مقدار بهینه خوبی پیدا می کند اما پیدا کردن هاپیر پارامتر مناسب برای این روش اندکی دشوار است. نمودار آخرین گام را در زیر مشاهده می کنیم.



روش بعدی که **rmsprop** است تقریباً مشابه با **adagrad** است که دارای مومنتم است. از این لحاظ همانطور که در **sgd** نیز مشاهده کرده بودیم زمانی که با مومنتم مدل را فیت می کنیم باعث می شود سریعتر همگرا شود پس میزان لرنینگ ریت را می توانیم پایینتر در نظر بگیریم. با لرنینگ ریت 40 و اپسیلون 0.01 و مومنتم 0.9 در 9 گام به مقدار $loss = 933.0111$ رسیدیم. نمودار زیر آخرین گام را نمایش می دهد.



آخرین اپتیمایزر نیز متعلق به ادم است. سخت ترین انتخاب پارامتر مربوط به این اپتیمایزر است که به علت تعداد بالای پارامترهای آن است در حالی که این اپتیمایزر یکی از بیشترین استفاده ها را در یادگیری ماشین و به خصوص شبکه های عصبی دارد. پس از چندین بار آزمون و خطا با مقدار لرنینگ ریت 0.9 و مومنتم اول یا **betha1** برابر با 0.9 و مومنتم دوم یا **betha2** برابر با 0.01 همچنین میزان اپسیلون 0.01 پس از 29 گام به مقدار خطای $loss = 934.0723$ رسیدیم که اندکی از حالت های قبلی بالاتر است.

به طور کلی به نظرم هابیر پارامترهای خوبی برای این روش پیدا نشده است وگرنه این اپتیمایزر یکی از بهترین اپتیمایزرها می باشد.

