

Lista de Exercícios #2 – Regressão Linear

1. Problema: Há um conjunto de treinamento com milhares de features.

Considerando um conjunto de treinamento com milhares de features, a técnica de regressão linear utilizada, seria a do Gradiente Descendente Estocástico (SGD). A técnica SGD foi a escolhida, por utilizar de forma aleatória um exemplo dentro do conjunto de treinamentos, a cada iteração, e a partir deste, realizar o cálculo do gradiente, a fim de buscar o mínimo global da superfície de erro.

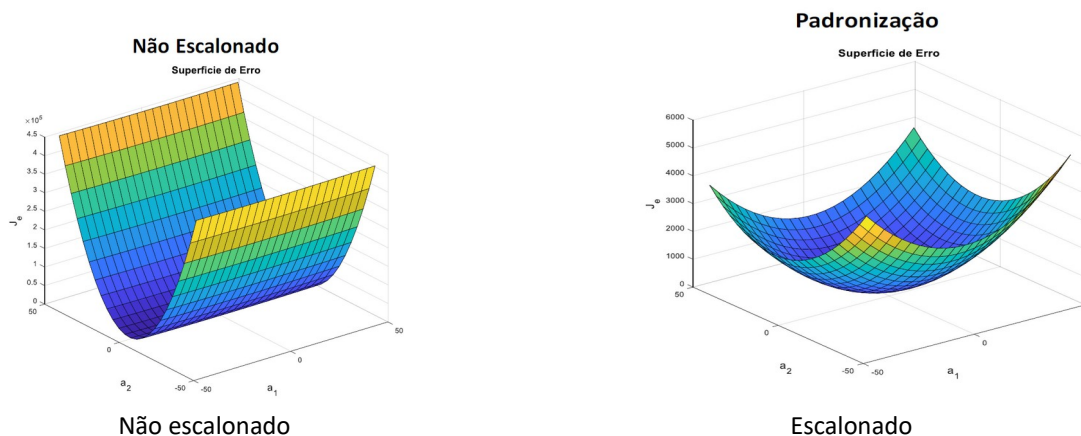
É o mais recomendado para casos que envolvam um grande número de features, pois não é necessária a utilização de todos os exemplos de treinamento a cada iteração. Por consequência, demanda menos recursos computacionais e é menos complexa a sua implantação.

2. Problema: features com escalas muito diferentes.

Para casos em que as features possuam escalas muito diferentes, todas as técnicas de regressão linear, baseadas no cálculo de distância euclidiana (ex: como as que utilizam superfície de erro, baseadas no erro quadrático médio), podem sofrer.

Estas técnicas baseadas em cálculo de distância, são influenciadas, pois as superfícies de erro se tornam “irregulares”, e ou, não convexas. Tal irregularidade, resulta em parâmetros de maiores magnitudes contribuindo mais para o cálculo do erro na superfície, comparado a parâmetros de menor magnitude. Como consequência, o desempenho do aprendizado é influenciado e o valor mínimo encontrado não é ótimo.

Para mitigar este problema, devem ser utilizadas técnicas de escalonamento de features, na fase de pré-processamento do conjunto de treinamento. Estas técnicas de escalonamento, visam condicionar todas as features em um mesmo intervalo de magnitudes, resultando na contribuição adequada de todas as features no processo de cálculo de erro, e consequentemente, na busca do valor mínimo global. Podem ser implementadas através do escalonamento por padronização ou normalização.



3. Problema: é utilizado GD em batelada. O erro aumenta constantemente.

O aumento do valor de erro, ao utilizar o GD em batelada, pode ser explicado pela utilização de um valor inadequado para o passo de aprendizado (α).

Para resolver tal problema, deve-se encontrar um valor ótimo para α , utilizando métodos de ajuste de α , como o de variação programada ou adaptativa.

Ajuste de α com variação programada: neste caso, o valor de α decresce em determinado valor, ao se atingir determinado número de iterações ou épocas. É feito ajuste inicial nos parâmetros, e em seguida, os valores de α encontrados são aplicados a todos. Ex: $\alpha = 0.1$ por 1000 iterações, em seguida, $\alpha = 0.05$ por outras 1000 iterações. Momentum, decaimento por etapas e etc.

Ajuste de α com variação adaptativa: o valor de α é ajustado dinamicamente, através de valores obtidos do próprio modelo de aprendizado. Este ajuste, permite que os parâmetros sejam atualizados individualmente, sem que seja necessário, a priori, ajuste inicial. Ex: Adam, RMSprop e etc.

4. Problema: dado os algoritmos de GD, batch, mini-batch e estocástico.

O algoritmo que possui o melhor desempenho, para se alcançar a vizinhança da solução ótima, é o estocástico. Seu melhor desempenho é justificado, por utilizar apenas um exemplo do conjunto de treinamento, a cada iteração, para o cálculo do gradiente.

O algoritmo que realmente converge, é o batch. Sua convergência ocorre porque este utiliza todos os exemplos do conjunto de treinamento a cada iteração/época. Sendo assim, o valor encontrado é realmente o mínimo global.

Para que os demais algoritmos também converjam (estocástico e mini-batch), podem ser utilizados ajustes que permitam a variação do passo de aprendizagem (α), como o ajuste de α com variação programada ou adaptativa (descritos acima).

