Regression

Linear Regression

Q. How does a linear regression model make a prediction?
A. By computing weighted sum of features plus a constant term called the bias/intercept

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots \theta_n x_n$$

$\hat{y}$: prediction; θ_n: parameter/weight; x_n: n^{th} feature

$$\hat{y} = h_\theta(\bar{x}) = \bar{\theta} \cdot \bar{x} \quad \# \text{ vectorized form}$$

Q. What are some cost functions that can be used to train a linear regression model?
A. These cost functions can be minimized to train a linear regression model:

- MSE

$$MSE = \frac{1}{m} \sum^{m} (h(x) - y)^2$$

- RMSE

$$RMSE = \sqrt{MSE}$$

- MAE

$$MAE = \frac{1}{m} \sum^{m} |h(x) - y|$$

Q. What is the normal equation?

A. The normal equation is a closed form solution a.k.a. a mathematical way of computing the parameters if a linear regression problem

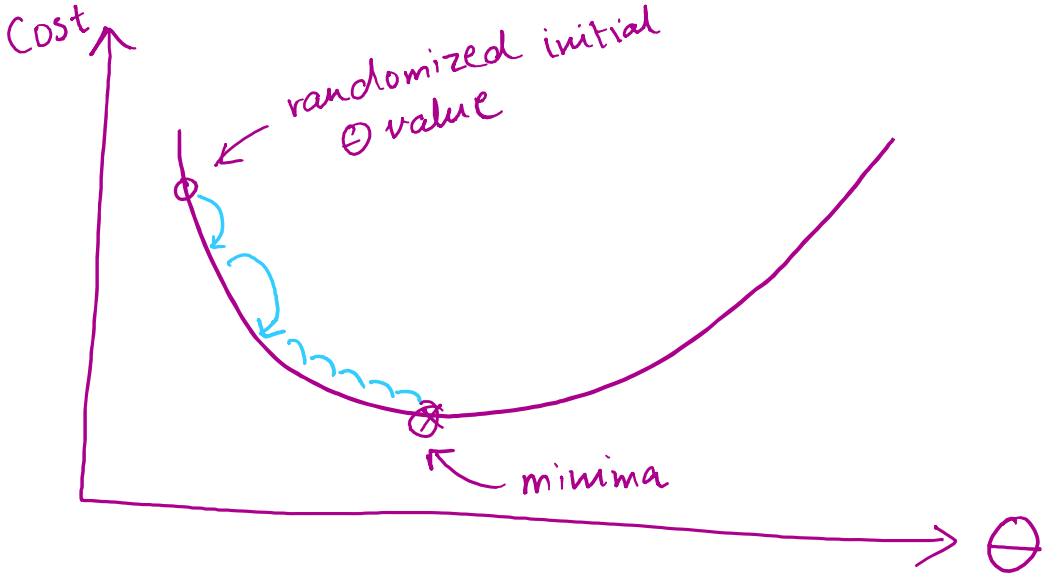
$$\hat{\theta} = (X^T X)^{-1} X^T y$$

Q. What is the computational complexity of linear regression (in sklearn)?
A. O(n^2), where n is the number of features. Making predictions is O(n) operation.

Gradient Descent

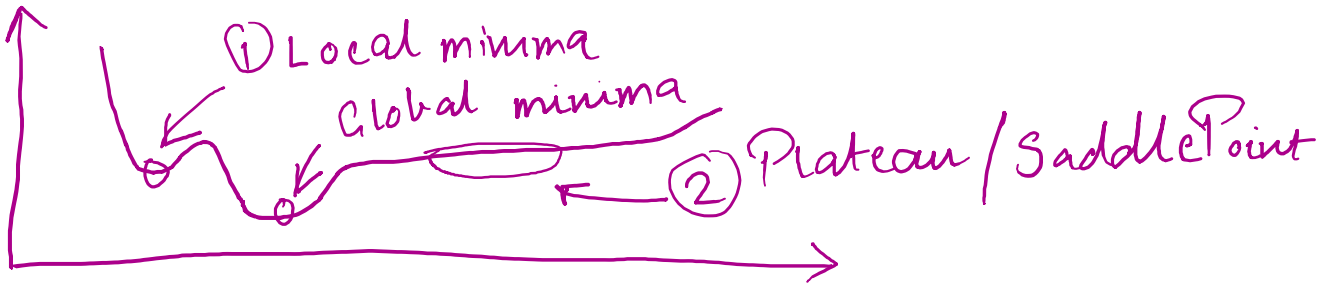
Q. Describe, in very simple terms, how gradient descent works?
A. The general idea of gradient descent is to tweak parameters gradually with the aim of minimizing the cost function:

- Initialize theta randomly or with 0s
- Tweak values in theta in steps such that the cost function is reduced
- Continue till the algorithm converges – i.e. a minima is reached



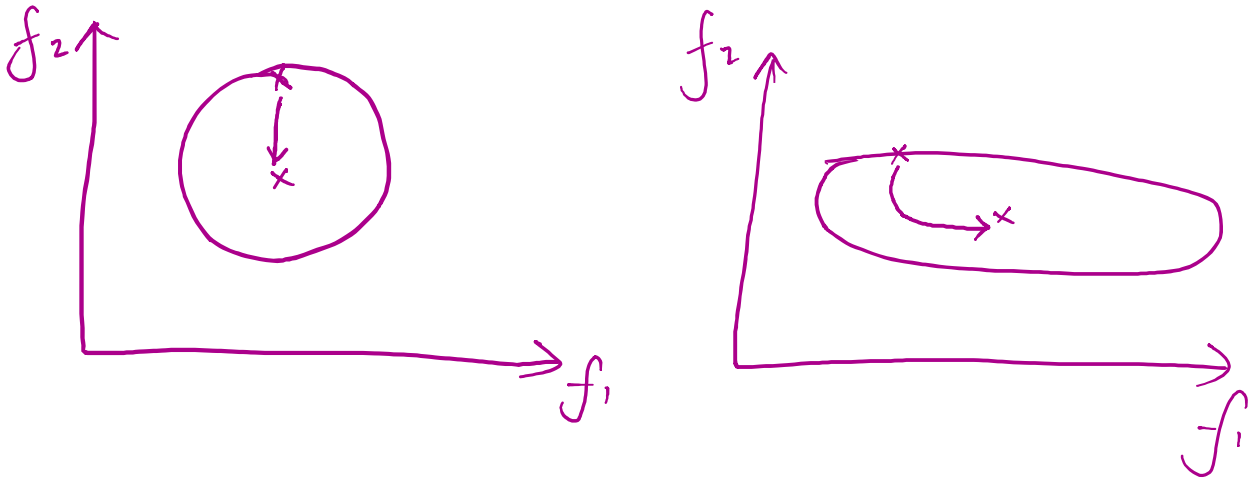
Q. What is the learning rate? Why is it important?
A. The learning rate is a hyperparameter that determines the step size the algorithm takes in an iteration to tweak the value of theta. If the learning rate is too small, the model will take too long to converge. If the learning rate is too high, the model may never converge and the value may diverge.

Q. What are 2 common challenges with gradient descent?
A.



Q. Why do we not face the challenge of a local minima in MSE for linear regression?
A. MSE cost function for linear regression is a convex function, which means that there is only a global minima and no local minima.

Q. Describe the problem of differing scales of features?
A. If feature 1 is on a scale of 1-100 while feature 2 is on a scale of 0-1, then this may cause the learning to take very long. We can use Sklearn's StandardScaler.



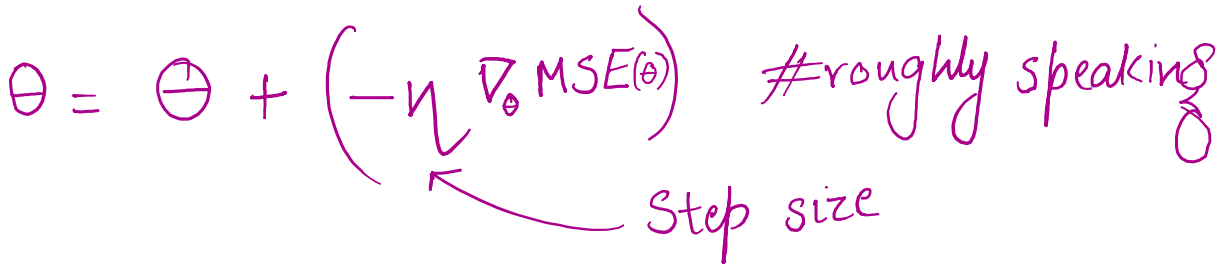
Batch Gradient Descent

Q. How do you compute the 'direction' in which the value of theta will be changed by the update algorithm?
A. By computing the derivative/slope/gradient of the cost function. The partial derivatives of the cost function are computed.

$$\frac{\partial}{\partial \theta_j} MSE(\theta) = \frac{2}{m} \sum^{m} \left(\theta^T x^{(i)} - y^{(i)}\right) x_j^{(i)}$$

or

$$\nabla_{\bar{\theta}} MSE(\bar{\theta}) = \begin{bmatrix} \frac{\partial}{\partial \theta_0} MSE \\ \frac{\partial}{\partial \theta_1} MSE \\ \vdots \\ \frac{\partial}{\partial \theta_n} MSE \end{bmatrix} = \frac{2}{m} \bar{X}^T (\bar{X}\theta - \bar{y})$$



Q. What is batch gradient descent?
A. When the gradient of the cost function is computed over the whole dataset X, it is called batch gradient descent aka full gradient descent.

Q. What is an advantage and disadvantage of batch gradient descent?
A. Advantage: It scales well with number of features as compared to Normal Equation or SVD decomposition
Disadvantage: It scales poorly with number of examples in the dataset

Q. What is a good way to set the number of iterations for which to run the gradient descent algorithm?
A. A good way is to set a high number of iterations but set a tolerance e. When the absolute value of the step_size*gradient becomes smaller than e, we stop the algorithm from updating.

Q. Describe the tolerance-convergence rate trade-off?
A. If the tolerance is too small, the algorithm will take a very long time to converge. If the tolerance is too large, the algorithm will stop quicker but then the solution will be further from the optimal solution. It takes O(1/e) time to converge. So if we set the tolerance to O(1/(e/10)) = O(10/e) it will take 10x longer.

Stochastic Gradient Descent

Q. What is SGD? What is its advantage over Batch GD?
A. SGD (Stochastic==random) picks a random instance at every step and computes gradients based only on that instance. This makes it much faster and a much more practical approach as compared to BGD.
The irregularity of SGD also helps the cost function in jumping out of a local minima.

Q. What is a disadvantage of using SGD?
A. Due to the random nature of SGD the cost function keeps bouncing around. Even when the algorithm stops, the solution reached is good but not optimal. The cost only decreases on average.

Q. How can the problem of SGD's erratic bouncing around be mitigated?
A. Gradually decrease learning rate using a learning schedule

Mini-Batch Gradient Descent

Q. What is the advantage of MBGD?
A. Best of both worlds. SGD but on a batch and can leverage on hardware optimization using GPUs.

Polynomial Regression

Q. What is Polynomial Regression?
A. A polynomial regression model is a handy tool to use when the data is more complex than what can be handled by a simple linear model. Powers are added to features to generate new features, then a linear model is trained on this extended set of features.

Q. Will adding more data help in the case of underfitting?
A. No, that generally helps in the case of overfitting. In this case a more complex model is required.

Q. What are bias, variance and irreducible error?
A. A model's generalization error can be expressed as the sum of three different errors:

- Bias: Error due to wrong assumptions eg: data is linear when it is actually quadratic
- Variance: Error due to model's hyper-sensitivity to data i.e. the model's complexity causing overfitting
- Irreducible Error: Error due to the noisiness of the data itself. Clean up the data to reduce

So, increasing the model's complexity will increase variance and reduce bias. The converse is true too.

Ridge Regression

Q. What is ridge regression?
A. By adding a penalty to the cost function, it is ensured that the weights/parameters are as small as possible, thereby reducing overfitting. Alpha=0 means a simple linear regression model. A very large alpha means that all the weights end up being very close to 0.

$$\text{Cost } f(x) = MSE(\theta) + \alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$$

Q. Why is it important to scale the input features before performing ridge regression?
A. Ridge regression is very sensitive to the scale of the input features. Therefore, it is very important to scale the features. This is true for most regularized models.

Q. What is LASSO Regression?
A. Leas Absolute Shrinkage and Selection Operator Regression. Uses L1 norm of the weight vector.

$$J\theta = MSE(\theta) + \alpha \sum_{i=1}^{n} |\theta_i|$$

Q. What is unique about Lasso?
A. It tends to completely eliminate the weights of the least important features by setting them to 0. It automatically performs feature selection and outputs a sparse model (few non-zero weights)

Elastic Net

Q. What is elastic net?
A. Combination of lasso and ridge

$$J(\theta) = MSE(\theta) + r\alpha \sum_{i}^{n} |\theta_i| + \frac{1-r}{2} \alpha \sum_{i}^{n} \theta_i^2$$

(r: ratio of lasso to ridge)

Q. What is early stopping?
A. Stopping the learning when validation error reaches a minimum. In the case of SGD, MBGD, validation curve might not be so smooth. In that case, we can let the validation loss increase a bit and then roll back to a model saved at the min.

Logistic Regression

Q. What is Logistic Regression?
A. A form of regression used for classification by using a logistic function. Like linear regression the model computes weighted sum of input features + bias term, but instead of outputting this sum it outputs the logistic of this result (probability).

$$\hat{p} = h_{\bar{\theta}}(\bar{x}) = \sigma(x^T \bar{\theta})$$

where $\sigma(t) = \frac{1}{1+\exp(-t)}$ } logistic function

$$\hat{y} = \begin{cases} 0 & \text{if } p < 0.5 \\ 1 & \text{if } p \geq 0.5 \end{cases}$$

Q. What is the cost function used to train a logistic regression model?
A.

$$J(\theta) = -\frac{1}{m} \sum_{1}^{m} \left[y^{(i)} \log(\hat{p}^{(i)}) + (1-y^{(i)}) \log(1-\hat{p}^{(i)}) \right]$$

Logistic Cost function partial derivatives

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(\sigma(\theta^T x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

Softmax Regression

Q. What is softmax regression?
A. Softmax regression is a generalized version of logistic regression for multi-class classification without having to combine several binary classifiers. It is aka Multinomial Logistic Regression.

Q. How is softmax computed?

$$\hat{p}_k = \sigma(s(x))_k = \frac{\exp(s_k(x))}{\sum_{j=1}^{K} \exp(s_j(x))}$$

k = k^{th} class

K = total no. of classes

$$s_k(x) = x^T \theta^{(k)}$$

each class has its own dedicated parameter vector $\theta^{(k)}$

- $s(x)$: vector containing the scores of each class for instance x
- $\sigma(s(x))_k$: probability that x belongs to class k

Prediction

$$\hat{y} = \underset{k}{\operatorname{argmax}}\ \sigma(s(x))_k = \operatorname{argmax}\ s_k(x)$$

$$= \operatorname{argmax}\left((\theta^{(k)})^T x\right)$$

returns a value k that maximizes the estimated probability

Q. What is the cost function used for softmax regression?
A. Cross-entropy function

$$J(\theta) = \frac{-1}{m}\sum_{k=1}^{m} y_k^{(i)} \log\left(\hat{p}_k^{(i)}\right)$$

$y_k^{(i)}$ → target probability that instance i belongs to class k
usually = 0 or 1

Gradient Vector

$$\nabla_{\theta^{(k)}} J(\theta) = \frac{1}{m_i}\sum_{i=1}^{m} \left(\hat{p}_k^{(i)} - y_k^{(i)}\right) x^{(i)}$$

Questions about Regression

Q. What Linear Regression training algorithm can you use if you have a training set with millions of features?
A. Gradient Descent – either SGD, MBSGD or maybe even BGD if the dataset fits in memory

Q. Suppose the features in your training set have very different scales. What algorithms might suffer from this, and how? What can you do about it?
A. Any algorithm that regularizes by penalising magnitude of features – LASSO, Ridge, Elastic Net. Use a scaler like min-max scaler.

Q. Can Gradient Descent get stuck in a local minimum when training a Logistic Regression model?
A. No, logistic regression cost function is convex and only has one minima.

Q. Do all Gradient Descent algorithms lead to the same model provided you let them run long enough?
A. Possibly similar but no guarantee of exactly the same.

Q. Suppose you use Batch Gradient Descent and you plot the validation error at every epoch. If you notice that the validation error consistently goes up, what is likely going on? How can you fix this?
A. The model is overfitting. Reduce complexity of model by reducing degree of polynomial features. Regularize the model by adding l1 or l2 penalty.

Q. Is it a good idea to stop Mini-batch Gradient Descent immediately when the validation error goes up?
A. No. Let it increase a bit and then roll back to value at minima.

Q. Which Gradient Descent algorithm (among those we discussed) will reach the vicinity of the optimal solution the fastest? Which will actually converge? How can you make the others converge as well?
A. Stochastic Gradient Descent will be the fastest. All will generally converge but Batch Gradient Descent generally always will. Using a good learning schedule convergence can be ensured.

Q. Suppose you are using Polynomial Regression. You plot the learning curves and you notice that there is a large gap between the training error and the validation error. What is happening? What are three ways to solve this?
A. Underfitting. Increase complexity of model by increase degrees of the polynomial features. Increase the amount of data and the richness/quality of the data. Reduce the regularization alpha.

9. Suppose you are using Ridge Regression and you notice that the training error and the validation error are almost equal and fairly high. Would you say that the model suffers from high bias or high variance? Should you increase the regularization hyperparameter α or reduce it?

Q. Why would you want to use:
• Ridge Regression instead of plain Linear Regression (i.e., without any regularization)?
    A. Because we want to regularize the model to prevent overfitting

• Lasso instead of Ridge Regression?
    A. For the auto-feature selection that Lasso does by changing weights to 0 for unimportant features

• Elastic Net instead of Lasso?
    A. Lasso can sometimes be a bit erratic especially when features are strongly correlated.

Q. Suppose you want to classify pictures as outdoor/indoor and daytime/nighttime.
Should you implement two Logistic Regression classifiers or one Softmax Regression classifier?
A. They seem to be two separate problems so its better to train two separate logistic regression models since the classes won't be mutually exclusive.