

NEURAL NETWORKS AND DEEP LEARNING CST 395 CS 5TH SEMESTER HONORS COURSE- Dr Binu V P, 9847390760

CS 5th Semester Honors course for the Computer Science at KTU- Dr Binu V P

Linear regression



September 30, 2022

Our definition of a machine learning algorithm as an algorithm that is capable of improving a computer program's performance at some task via experience is somewhat abstract. To make this more concrete, we present an example of a simple machine learning algorithm: **linear regression**

As the name implies, linear regression solves a regression problem. In other words, the goal is to build a system that can take a vector $x \in R_n$ as input and predict the value of a scalar $y \in R$ as its output. In the case of linear regression, the output is a linear function of the input. Let \hat{y} be the value that our model predicts y should take on. We define the output to be

$$\hat{y} = w^T x$$

where $w \in R^n$ is a vector of parameters. Parameters are values that control the behavior of the system. In this case, w_i is the coefficient that we multiply by feature x_i before summing up the contributions from all the features. We can think of w as a set of weights that determine how each feature affects the prediction. If a feature x_i receives a positive weight w_i , then increasing the value of that feature increases the value of our prediction \hat{y} . If a feature receives a negative weight, then increasing the value of that feature decreases the value of our prediction. If a feature's weight is large in magnitude, then it has a large effect on the prediction. If a feature's weight is zero, it has no effect on the prediction.

We thus have a definition of our task T : to predict y from x by outputting $\hat{y} = w^T x$.

Next we need a definition of our performance measure, P . Suppose that we have a design matrix of m example inputs that we will not use for training, only for evaluating how well the model performs. We also have a vector of regression targets providing the correct value of y for each of these examples. Because this dataset will only be used for

evaluation, we call it the test set. We refer to the design matrix of inputs as $X^{(test)}$ and the vector of regression targets as $y^{(test)}$.

One way of measuring the performance of the model is to compute the mean squared error of the model on the test set. If $\hat{y}^{(test)}$ gives the predictions of the model on the test set, then the mean squared error is given by

$$MSE_{test} = \frac{1}{m} \sum_i (\hat{y}^{(test)} - y^{(test)})_i^2$$

Intuitively, one can see that this error measure decreases to 0 when

$\hat{y}^{(test)} = y^{(test)}$. We can also see that

$$MSE_{test} = \frac{1}{m} \|\hat{y}^{(test)} - y^{(test)}\|_2^2$$

so the error increases whenever the Euclidean distance between the predictions and the targets increases.

To make a machine learning algorithm, we need to design an algorithm that will improve the weights w in a way that reduces MSE_{test} when the algorithm is allowed to gain experience by observing a training set $X^{(train)}, y^{(train)}$. One intuitive way of doing this is just to minimize the mean squared error on the training set, MSE_{train}

To minimize MSE_{train} , we can simply solve for where its gradient is 0:

$$\nabla_w MSE_{train} = 0$$

$$\Rightarrow \nabla_w \frac{1}{m} \|\hat{y}^{(train)} - y^{(train)}\|_2^2 = 0$$

$$\Rightarrow \nabla_w \frac{1}{m} \|X^{(train)} w - y^{(train)}\|_2^2 = 0$$

$$\Rightarrow \frac{1}{m} \nabla_w (X^{(train)} w - y^{(train)})^T (X^{(train)} w - y^{(train)}) = 0$$

$$\Rightarrow \nabla_w (X^{(train)} w - y^{(train)})^T (X^{(train)} w - y^{(train)}) = 0$$

$$\Rightarrow \nabla_w (w^T X^{(train)^T} - y^{(train)^T}) (X^{(train)} w - y^{(train)}) = 0$$

$$\Rightarrow \nabla_w (w^T X^{(train)^T} X^{(train)} w - 2w^T X^{(train)^T} y^{(train)} + y^{(train)^T} y^{(train)}) = 0$$

$$\Rightarrow (2w^T X^{(train)^T} X^{(train)} - 2X^{(train)^T} y^{(train)}) = 0$$

$$w = (X^{(train)^T} X^{(train)})^{-1} X^{(train)^T} y^{(train)}$$

The system of equations whose solution is given by equation above is known as the **normal equations**. Evaluating equation constitutes a simple learning algorithm. See the figure below for an example of the linear regression learning algorithm in action

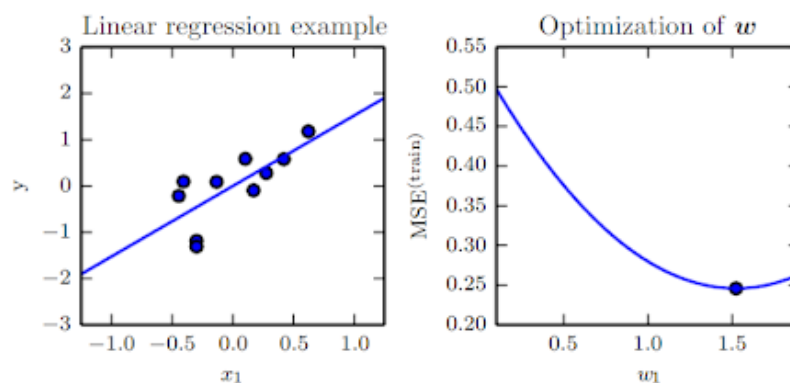


Figure 5.1: A linear regression problem, with a training set consisting of ten data points, each containing one feature. Because there is only one feature, the weight vector \mathbf{w} contains only a single parameter to learn, w_1 . (Left) Observe that linear regression learns to set w_1 such that the line $y = w_1 x$ comes as close as possible to passing through all the training points. (Right) The plotted point indicates the value of w_1 found by the normal equations, which we can see minimizes the mean squared error on the training set.

It is worth noting that the term linear regression is often used to refer to a slightly more sophisticated model with one additional parameter—an intercept term b . In this model $\hat{y} = w^T x + b$. So the mapping from parameters to predictions is still a linear function but the mapping from features to predictions is now an affine function. This extension to affine functions means that the plot of the model's predictions still looks like a line, but it need not pass through the origin. Instead of adding the bias parameter b , one can continue to use the model with only weights but augment x with an extra entry that is always set to 1. The weight corresponding to the extra 1 entry plays the role of the bias parameter. We will frequently use the term “linear” when referring to affine functions throughout.

The intercept term b is often called the bias parameter of the affine transformation. This terminology derives from the point of view that the output of the transformation is biased toward being b in the absence of any input. This term is different from the idea of a statistical bias, in which a statistical estimation algorithm's expected estimate of a quantity is not equal to the true quantity.

Linear regression is of course an extremely simple and limited learning algorithm, but it provides an example of how a learning algorithm can work.



Popular posts from this blog

NEURAL NETWORKS AND DEEP LEARNING CST 395 CS 5TH SEMESTER HONORS COURSE NOTES - Dr Binu V P, 9847390760

October 03, 2022

About Me Syllabus Question Paper Dec 2022 Module 1 (Basics of Machine Learning)
Overview of Machine Learning Machine Learning Algorithm Linear Regression Capacity,
Overfitting and Underfitting Regularization Hyperparameters and Validation ...

[READ MORE](#)

Machine Learning Algorithm

October 01, 2022

A machine learning algorithm is an algorithm that is able to learn from data. But what do we mean by learning? Mitchell (1997) provides the definition "A computer program is said to learn from experience E with respect to some class of tasks T and ...

 [Powered by Blogger](#)

[READ MORE](#)

Theme images by [Michael Elkan](#)

Syllabus

October 01, 2022

Syllabus Module - 1 (Basics of Machine Learning) Machine Learning basics - Learning algorithms - Supervised, Unsupervised, Reinforcement, overfitting, Underfitting, Hyper parameters and Validation sets, Estimators -Bias and Variance. Challenge: ...

[READ MORE](#)

DR.BINU V P-9847390760

Associate Professor and Head Dept of
Computer Science-IHRD

Karunagappally. Love to share the
thoughts and views .I play lot of Games
and basically an Athlet in my school and
college days.I never keep my studies as
a second option.Stood first In MTech
and BTech.I did my resear ...

[VISIT PROFILE](#)

Archive



[Report Abuse](#)