Linear Regression and Gradient Descent an Introduction to Supervised Learning

1st Laurenz Hundgeburth
Systems Engineering
Laurenz.Hundgeburth@edu.fh-kaernten.ac.at

2nd Gregor Fritz
Systems Engineering
edufrigre001@fh-kaernten.at

***Abstract*—This paper aims to explain the concept of supervised learning, by introducing the reader carefully to linear regression and gradient descent.**

***Keywords*—machine learning, supervised learning, linear regression, gradient descent, optimization**

I. Introduction

Previously machines were only as intelligent as the person who programmed them, but nowadays artificial intelligence and machine learning offer completely new and interesting ways of enriching computers capabilities. With them computers are able to learn to interpret data and find the hidden rules and laws inherent to them.

As an introduction to this topic, it is helpful to remind ourselves of how we learned to recognize numbers when we were at school. We were never told precise specifications or complex rules on how the numbers are comprised. Countless times the different numbers were shown to us and labeled with their corresponding meaning. Over time we just "learned" the different digits and became better at distinguishing them.

Linear regression is the simplest and most basic form of machine learning. With this paper we aim to explain linear regression in a simple and compact form, providing students with a short introduction to this topic.

II. Machine learning and Supervised Learning

Machine learning is all about finding a mathematical model representing our training data. It is called "learning", because the process of finding the model resembles human learning. The general work-flow and idea is depicted in Fig. 1. [5]

We feed our training data, which is comprised of some features *X* and the corresponding labels *y*, into a machine learning algorithm like linear regression. The result is a model, or hypothesis [3], which represents our data. Given a new input *X*, we can now predict the output *y* using the trained model.

In unsupervised learning, we only provide input data, which we also call "features", and let the algorithm find some structures and rules by itself. In supervised learning we "label" the training data by additionally providing the "correct" output.

III. Model

Finding a good model representing our training data is the ultimate goal of machine learning. Having such a model allows us to make reasonable accurate predictions.

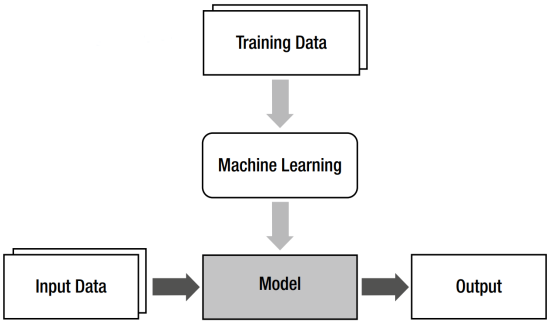

Fig. 1. Supervised Learning [5]

However, one should always keep in mind that the model is only a derived representation of our training data. The trained model may therefore be prone to erroneous predictions.

To solve regression problems we want to have a continuous output provided by a function $f : \mathbb{R}^n \to \mathbb{R}$ [1]. In the case of linear regression our hypothesis looks like equation (1), where Θ is a vector containing all the parameters of our model.

$$h_\Theta(X) = \Theta_0 + \Theta_1 \cdot x_1 \tag{1}$$

IV. Linear Regression

As the name implies, Linear Regression is an algorithm to solve regression problems, by finding a linear representation of our training data.

For example, the problem of predicting the resale value of a car in percent, according to its age in years.

In Fig. 2 you can see the resulting linear function (our hypothesis) representing our data.

Linear Regression is an iterative algorithm. We start by initializing our model parameters Θ randomly. On each iteration we then compute the performance of our model with a suitable cost function $J(\Theta)$. The next step is to improve the performance of our model by adjusting every parameter. If the algorithm is running correctly the cost function will decrease on every iteration until it converges to a local minimum.

V. Cost Function

The cost function $J(\Theta)$, also called the *objective function* or *criterion* [1], measures how well our hypothesis $h_\Theta(X)$ represents our training data. [3] There are many different cost functions. We decided to use the *Least Squares Method* shown

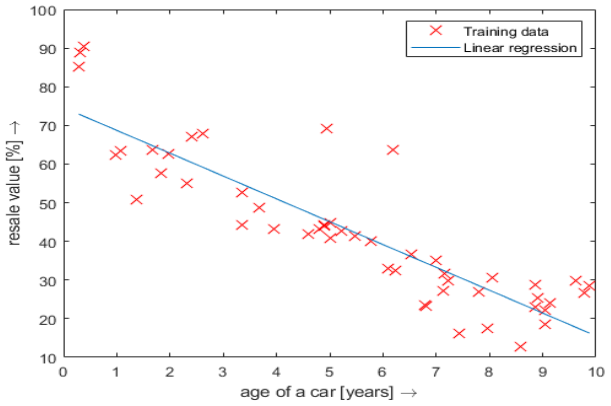

Fig. 2. Linear Regression Example

in equation (4), because it offers good accuracy with justifiable complexity.

$$J(\Theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\Theta(x^{(i)}) - y^{(i)}\right)^2 \tag{2}$$

This formula basically computes the average squared distance between each prediction $h_\Theta(x^{(i)})$ and the actual values $y^{(i)}$ by summing them up and divide them by the size m of the training data.

It is helpful to realize that the actual value of the cost function is of minor importance, as it is just a means to measure the performance of our hypothesis. In order to find suitable values for our model parameters, we have to know, how a change in every parameter is affecting the cost of our model. In mathematical terms, the derivatives of the cost function are crucial for the optimization step described in VI.

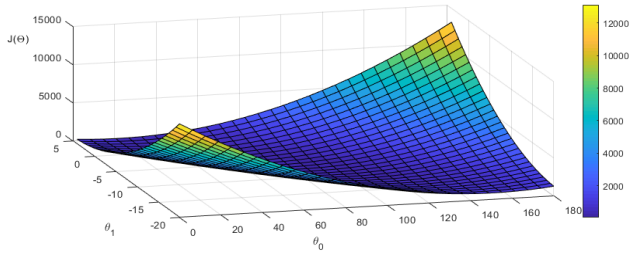

Fig. 3. Surface Plot of J(Θ)

VI. Optimization: Gradient Descent

As we mentioned in V, finding a good model boils down to an optimization problem. We need to find a global minimum of the cost function $J(\Theta)$. Finding such a minimum may be a trivial task for two dimensional problems, but when you have many features in your training data leading to a high dimensional optimization problem, you need to apply numerical methods like gradient descent (3).

$$\Theta_j := \Theta_j - \alpha\frac{\partial}{\partial\Theta_j}J(\Theta) \tag{3}$$

Gradient descent is an iterative process that loosely resembles descending a hill or mountain the fastest way possible by always taking a step in the direction of the steepest decline. This can be seen in (3), where the j^{th} parameter Θ_j is adjusted by the partial derivative of $J(\Theta)$ with respect to Θ_j - the gradient of $J(\Theta)$. We subtract the gradient, because we want to descent and the gradient points to the steepest ascent. α is called the learning rate and simply scales the step. On every iteration you have to compute the derivatives anew and repeat the adjusting process. Equation (4) shows how to compute the derivatives in our model when you set $x_0^{(i)} = 1$.

$$\Theta_j := \Theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\Theta(x^{(i)}) - y^{(i)}\right)\cdot x_j^{(i)} \tag{4}$$

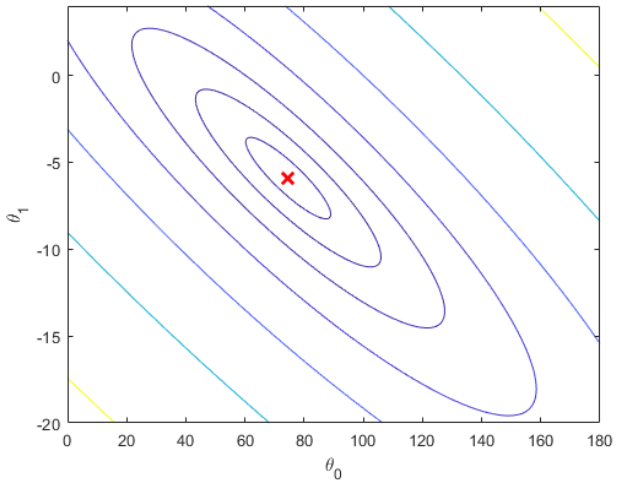

Fig. 4. Contour Plot of J(Θ)

VII. Conclusion

We saw how elegant gradient descent found the optimal solution to our regression problem by simply providing a suitable cost function and its derivatives. The beauty of machine learning lies in the fact that most advanced machine learning algorithms follow the same basic principle. We highly encourage the interested reader to either delve into this topic by following professor Andrew Ng's fantastic online course [3], or read one of the books we used in our preparation phase [1], [2] or [4].

References

[1] I. Goodfellow, Y. Bengio and A. Courville, Deep learning. Cambridge, Mass: The MIT Press, 2017.
[2] J. S. Marsland, Machine Learning An Algorithmic Perspective, 2nd ed. Boca Raton: Chapman & Hall/CRC, 2014.
[3] A. Ng, "Machine Learning — Coursera", Coursera, 2019. [Online]. Available: https://www.coursera.org/learn/machine-learning. [Accessed: 08- Jan- 2019].
[4] S. Russell and P. Norvig, Artificial intelligence A Modern Approach, 3rd ed. New Jersey: Pearson Education, 2010.
[5] P. Kim, MATLAB Deep Learning. Berkeley, CA: Apress, 2017.