Mathematics For AI

Madiba Hudson-Quansah

Contents

Chapter 1

Foundations

The main structure of the machine learning part of AI is as follows:

1. Identify the problem
2. Acquire the appropriate data
3. Create a hypothesis/ learning/ prediction / training function
4. Find the numerical values of weights
5. Create an error function
6. Decide on mathematical formulas
7. Find a way to search for minimizers
8. Use the backpropagation algorithm
9. Regularize a function

1.1 Vector Notation

Definition 1.1.1: Vector

A vector is a mathematical object that has both magnitude and direction. It is represented by an arrow with a starting point and an end point. The length of the arrow represents the magnitude of the vector, and the direction of the arrow represents the direction of the vector. Denoted by $\vec{v}$ or $\mathbf{v}$ which in vector space $\mathbb{R}^n$ represents:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$$

The transpose of a column vector $\mathbf{v}$ is denoted by $\mathbf{v}^t$ and is a row vector:

$$\mathbf{v}^t = \begin{pmatrix} v_1 & \dots & v_n \end{pmatrix}$$

Definition 1.1.2: Dot Product

The dot product of two vectors $\mathbf{v}$ and $\mathbf{w}$ is a scalar value denoted by $\mathbf{v} \cdot \mathbf{w}$ and is defined as:

$$\mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^{n} v_i w_i$$

The dot product is the same as the product of a row vector / transposed column vector and a column vector, for example for vectors $\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix}$, the dot product is:

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} &= \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} \\ &= a_1 b_1 + a_2 b_2 + a_3 b_3 + a_4 b_4 + a_5 b_5 \end{aligned}$$

Or

$$\begin{aligned} \mathbf{a}^t \mathbf{b} &= \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} \\ &= a_1 b_1 + a_2 b_2 + a_3 b_3 + a_4 b_4 + a_5 b_5 \end{aligned}$$

Moreover:

$$\| \mathbf{a} \|_{l^2}^2 = \mathbf{a}^t \mathbf{a} = a_1^2 + a_2^2 + a_3^2 + a_4^2 + a_5^2$$

1.2 Numerical vs Analytical Solutions

Definition 1.2.1: Numerical

Has to do with numbers. Are approximations of analytical solutions obtained by discretizing a continuous problem.

Definition 1.2.2: Analytical

Has to do with analysis. Are exact solutions to problems. Not all problems have analytical solutions.

1.3 Gradient Descent

Definition 1.3.1: Gradient Descent

An optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient.

1.4 Regression

Definition 1.4.1: Regression

A statistical method used to determine the strength and character of the relationship between one dependent variable and one or more independent variables.

1.5 Classification

Definition 1.5.1: Classification

A supervised learning approach that categorizes input data into one of a number of classes.

Chapter 2

Linear Models

2.1 Linear Regression

A linear regression model is a linear approach to modelling the relationship between a dependent variable and one or more independent variables. In using a linear model we make the assumption that the dependent variable / predicted value depends linearly on the independent variables / features.

2.1.1 Simple Linear Regression

In the case of simple linear regression, there is only one independent variable. This relationship can be expressed as:

$$y = \alpha + \beta x$$

Where y is the predicted value / dependent variable, α is the intercept, β is the slope, and x is the feature / independent variable.

2.1.1.1 Training Function

2.1.1.2 Loss Function

2.1.1.3 Optimization

2.1.2 Multi-Linear Regression

In the case of multi-linear regression, there are multiple independent variables. This relationship can be expressed as:

$$y = f(x_1, x_2, x_3, \ldots, x_n)$$

Where y is the predicted value, $x_1, x_2, x_3, \ldots, x_n$ are the features, and f is the function that maps the features to the predicted value.

2.1.2.1 Training Function

As with all linear models the assumption of linear dependence is made. This means that the predicted value is a linear combination of its features plus a bias term ω_0, giving the following training function:

$$y = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_n x_n$$

Where y is the predicted value, ω_0 is the bias term, $\omega_1, \omega_2, \ldots, \omega_n$ are the weights, and $x_1, x_2, \ldots, x_n$ are the features. In order to find the appropriate weights ω_n, they have to be learnt from the training data. This process is called training the model. Ergo a trained model is model that has decided on the appropriate weights to use.

Each weight ω is a scalar value multiple by a corresponding feature x, and represent the importance of a specific feature in determining the predicted value. This means the larger the weight the more important the feature is in determining

the predicted value, and vice versa. *Dead features* are features that have a weight of zero and do not affect the predicted value. This means for each row in the total amount of rows m the prediction of this model is:

$$\begin{aligned} y^1_{\text{predict}} &= \omega_0 + \omega_1 x^1_1 + \omega_2 x^1_2 + \ldots + \omega_n x^1_n \\ y^2_{\text{predict}} &= \omega_0 + \omega_1 x^2_1 + \omega_2 x^2_2 + \ldots + \omega_n x^2_n \\ &\vdots \\ y^m_{\text{predict}} &= \omega_0 + \omega_1 x^m_1 + \omega_2 x^m_2 + \ldots + \omega_n x^m_n \end{aligned}$$

Which can be expressed as:

$$\begin{pmatrix} y^1_{\text{predict}} \\ y^2_{\text{predict}} \\ \vdots \\ y^m_{\text{predict}} \end{pmatrix} = \omega_0 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + \omega_1 \begin{pmatrix} x^1_1 \\ x^2_2 \\ \vdots \\ x^m_n \end{pmatrix} + \ldots + \omega_n \begin{pmatrix} x^1_1 \\ x^2_2 \\ \vdots \\ x^m_n \end{pmatrix}$$

$$\begin{pmatrix} y^1_{\text{predict}} \\ y^2_{\text{predict}} \\ \vdots \\ y^m_{\text{predict}} \end{pmatrix} = \begin{pmatrix} 1 & x^1_1 & x^1_2 & \ldots & x^1_n \\ 1 & x^2_1 & x^2_2 & \ldots & x^2_n \\ \vdots & & & & \\ 1 & x^m_1 & x^m_2 & \ldots & x^m_n \end{pmatrix} \begin{pmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{pmatrix}$$

2.1.2.1.1 Parametric Models vs Non-Parametric Models

Definition 2.1.1: Parametric Model

A model that has parameters that are fixed in number, regardless of the size of the training data. For example:

$$y = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_n x_n$$

The number of parameters (ω) is fixed at $n + 1$. This means that the formula of training function is fixed ahead of training the model, and the model is trained to find the appropriate values of the parameters.

Definition 2.1.2: Non-Parametric Model

A model that does not specify the formula for the training function with it's parameters before training the model. This results in the number of parameters being dependent on the size of the training data, as such models adapt to the data and determines the required number of parameters during training.

Both parametric and non-parametric models have other parameters that are not weights called hyperparameters, that need to be tuned during training.

Definition 2.1.3: Hyperparameters

Parameters that are not weights, and are not learnt during training. They are set before training the model, and determine the behaviour of the model.

2.1.2.2 Loss Function

To determine the weights of the model using the training data we need to define and optimize a loss function.

Definition 2.1.4: Loss Function

Determines the error of the model, by measuring the difference between the predicted values generated by the model and the actual values in the training data.

For example assuming we trained a model with n features, giving us the weights $\omega_0, \omega_1, \ldots, \omega_n$, then the ith predicted values using the ith row of the training data would be:

$$y^i_{\text{predict}} = \omega_0 + \omega_1 x^i_1 + \omega_2 x^i_2 + \ldots + \omega_n x^i_n$$

However for the ith row of the training data the actual value is y^i_{true}. In determining the error between the predicted value and the actual value we can use the following approaches.

Absolute Value Distance - $\mid y_{\text{predict}} - y_{\text{true}} \mid$ derived from $\mid x \mid$

Squared Distance - $\mid y_{\text{predict}} - y_{\text{true}} \mid^2$ derived from $\mid x \mid^2$

In examining both functions $\mid x \mid$ and $\mid x \mid^2$ we notice that the squared distance function is smoother than the absolute value distance function. This means that the squared distance function is differentiable at all points, while the absolute value distance function is not differentiable at $x = 0$, this is called a *singularity*.

Other than the difference in *regularity* of both functions another consideration must be made, *if a number is large, then its square is even larger.* This means that the squared distance function is more sensitive to outliers in the data than the absolute value distance function.

Definition 2.1.5: Regularity

The property of a function that determines how smooth it is. A function is regular if it is differentiable at all points.

2.1.2.2.1 Functions with Singularities

Definition 2.1.6: Singularity

A point at which a function is not differentiable.

Generally graphs of differential functions do not have sharp corners, cusps, or vertical tangents. This is because at that point you can draw two different tangents lines to the graph of the function at that point giving two different slopes at that point. This discontinuity in the slope of the tangent creates a problem for methods that rely on evaluating the derivative of the function such as the gradient descent algorithm.

2.1.2.2.2 Mean Squared Error

For linear regression models the most common loss function is the mean squared error function.

Definition 2.1.7: Mean Squared Error

The average of the squared differences between the predicted values and the actual values in the training data. Defined as:

$$\text{Mean Squared Error} = \frac{1}{m} \sum_{i=1}^{m} \mid y^i_{\text{predict}} - y^i_{\text{true}} \mid^2$$

Where m is the number of rows. And in linear algebra notation:

$$\text{Mean Squared Error} = \frac{1}{m} \left(\vec{y}_{\text{predict}} - \vec{y}_{\text{true}}\right)^t \left(\vec{y}_{\text{predict}} - \vec{y}_{\text{true}}\right) = \frac{1}{m} \parallel \vec{y}_{\text{predict}} - \vec{y}_{\text{true}} \parallel^2_{l^2}$$

Where l^2 denotes the l^2 norm of the vector which by definition is the $\sqrt{\text{sum of squares of its components}}$

2.1.2.3 Optimization

2.1.3 Polynomial Regression

Chapter 3

Decision Trees

3.1 Decision Tree Regressors

Chapter 4

Regularization

4.1 Regularization of Regression Models

Chapter 5

Evaluation

5.1 Evaluating Regression Models