Using neural networks for classification and regression problems

Eina B. Jørgensen, Anna Lina P. Sjur and Jan-Adrian H. Kallmyr

November 1, 2019

Abstract

1 Introduction

2 Theory

2.1 Logistic Regression

Classification problems aim to predict the behaviour of a given object, and look for patterns based on discrete variables (i.e categories). Logistic regression can be used to solve such problems, commonly by the use of variables with binary outcomes such as true/false, positive/negative, success/failiure etc., or in the specific credit card case: *risky/non-risky*

As opposed to linear regression, the equation one gets as a result of minimization of of the cost function by $\hat{\beta}$ using logistic regression, is non-linear, and is solved using minimization algorithms called *gradient descent methods*.

When predicting the the output classes in which an object belongs, the prediction is based on the design matrix $\hat{\mathbf{X}} \in \mathbb{R}^{n \times p}$ that contain n samples that each carry p features.

A distinction is made between *hard classification* - deterministically determine the variable to a cathegory, and *soft classification* - determines the probability that a given variable belongs in a certain cathegory. The latter is favorable in many cases, and logistic regression is the most used example og this type of classifier.

When using logistic regression, the probability that a given data point x_i belongs in a cathegory y_i is given by the Sigmoid-function (or logistic function):

$$\begin{aligned} p(t) &= \frac{1}{1+e^{-t}} = \frac{e^t}{1+e^t} \\ 1 - p(t) &= p(-t) \end{aligned} \tag{1}$$

Assuming a binary classification problem, i.e. y_i can be either 0 or 1, and a set of predictors $\hat{\beta}$ the Sigmoid function (1) gives the probabilities with relation:

$$p(y_i = 0|x_i, \hat{\beta}) = 1 - p(y_i = 1|x_i, \hat{\beta})$$

The total likelihood for all possible outcomes $\mathcal{D} = \{(y_i, x_i)\}$ is used in the Maximum Likelihood Estimation (MLE), aiming at maximizing the log/likelihood funciton (2). The likelihood function can be expressed with $\mathcal{D}$:

$$P(\mathcal{D}|\hat{\beta}) = \prod_{i=1}^{n} \Big[p(y_i = 1|x_i, \hat{\beta})\Big]^{y_i} \Big[1 - p(y_i = 0|x_i, \hat{\beta})\Big]^{1-y_i}$$

And the log/likelihood function is then:

$$\begin{aligned} P_{\log}(\hat{\beta}) = \sum_{i=1}^{n} \Big(& y_i \log \Big[p(y_i = 1|x_i, \hat{\beta})\Big] \\ & + (1 - y_i) \log[1 - p(y_i = 0|x_i, \hat{\beta})]\Big) \end{aligned} \tag{2}$$

The cost/error-function $\mathcal{C}$ (also called cross-entropy in statistics) is the negative of the log/likelihood. Maximizing $P_{\log}$ is thus the same as minimizing the cost function. The cost funciton is:

$$\begin{aligned}&\mathcal{C}(\hat{\beta}) = -P_{\log}(\hat{\beta}) = \\ &-\sum_{i=1}^{n} \Big(y_i \log \Big[p(y_i = 1|x_i, \hat{\beta}) \Big] \\ &\quad + (1 - y_i) \log[1 - p(y_i = 0|x_i, \hat{\beta})] \Big)\end{aligned} \tag{3}$$

Finding the parameters $\hat{\beta}$ that minimize the cost function is then done through derivation. Defining the vector $\hat{y}$ containing n elements y_i, the $n \times p$ matrix $\hat{X}$ containing the x_i elements, and the vector $\hat{p}$ that is the fittet probabilities $p(y_i|x_i, \hat{\beta})$, the first derivative of $\mathcal{C}$ is

$$\frac{\partial \mathcal{C}(\hat{\beta})}{\partial \hat{\beta}} = -\hat{X}^T(\hat{y} - \hat{p}) \tag{4}$$

This gives rise to set of linear equations, where the aim is to solve the system for $\hat{\beta}$. By introduction of a diagonal matrix $\hat{W}$ with diagonal elements $p(y_i|x_i, \hat{\beta}) \cdot (1 - p(y_i|x_i, \hat{\beta}))$ the second derivative is:

$$\frac{\partial^2 \mathcal{C}(\hat{\beta})}{\partial \hat{\beta} \partial \hat{\beta}^T} = \hat{X}^T \hat{W} \hat{X} \tag{5}$$

With $\hat{x} = [1, x_1, x_2, ..., x_p]$ and p predictors $\hat{\beta} = [\beta_0, \beta_1, \beta_2, ..., \beta_p]$ the ration between likelihoods of outcome is:

$$\log \frac{p(\hat{\beta}\hat{x})}{1 - p(\hat{\beta}\hat{x})} = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p \tag{6}$$

and $p(\hat{\beta}\hat{x})$ defined by:

$$p(\hat{\beta}\hat{x}) = \frac{e^{\beta_0 + \beta_1 x_1 + ... + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + ... + \beta_p x_p}} \tag{7}$$

2.2 Gradient Descent Methods

Matrise-eksempel

$$A = \begin{bmatrix} b_1 & c_1 & 0 & \dots & \dots & 0 \\ a_1 & b_2 & c_2 & 0 & \dots & 0 \\ 0 & a_2 & b_3 & c_3 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \ddots & a_{n-2} & b_{n-1} & c_{n-1} \\ 0 & \dots & \dots & 0 & a_{n-1} & b_n \end{bmatrix},$$

3 Results

Figure 1

4 Discussion

5 Conclusion

References