Multiple linear regression with Neural Networks in Tensorflow 2 and Pytorch

Davide Liso

1 Introduction

The purpose of this short paper is to use two neural networks (using Tensorflow 2 and Pytorch) to perform a multiple linear regression having the real consumption as dependent variable and as explanatory variables other macroeconomic variables listed below:

- gdp: Real gross domestic product
- invest: Real investment by private sector
- government: Real government expenditures
- dpi: Real disposable personal income
- cpi: Consumer price index
- m1: Nominal money stock
- tbill: Quarterly average of month end 90 day treasury bill rate
- unemp: Unemployment rate
- population
- inflation: Inflation rate
- interest: Ex-post real interest rate

The dataset contains the historical series of the above-mentioned US macroeconomic variables from 1950 to 2000. Regarding the preprocessing of the dataset, the first difference of all variables was calculated.Then, to eliminate the multicollinearity problem, only those variables with the variance inflation factor < 4 were selected.

feature	VIF
gdp	5.450997
dpi	2.476818
cpi	1.864907
unemp	1.714120
invest	2.709847
inflation	18.677120
interest	18.026460

For each column, values below the 15th percentile and above the 85th percentile were removed and replaced with the mean. Finally, the variables have been standardized to avoid the problem of the exploding gradient, since the implemented neural networks do not have an activation function (i.e. linear activation function)
I have used the SDG as optimizer, a constant learning rate to 1e-2 and MSE as loss for 200 epochs. The network architecture is shown below.

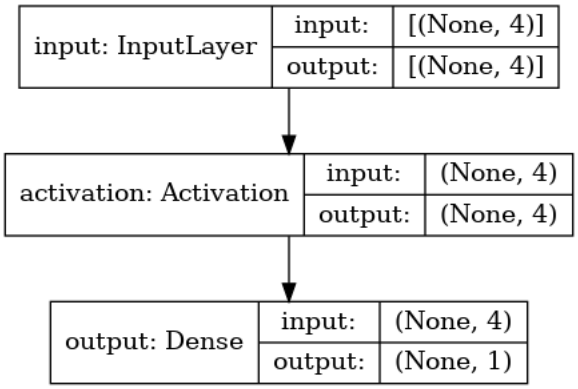

Figure 1: Network architecture is the same for both Tensorflow 2 and Pytorch

2 Results

2.1 Tensorflow 2

The multiple linear regression coefficients found by the neural network are as follows:

dpi	cpi	unemp	invest	bias
0.281196	0.142998	-0.143651	-0.0364	0.00368

The bias term is the constant in the linear regression model, while the graph below shows the epoch/loss

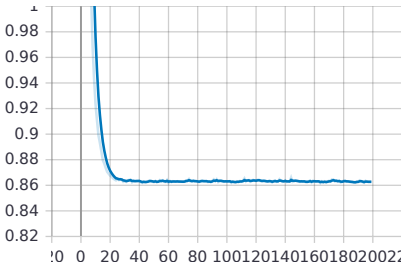

Figure 2: Epoch / Loss (Tensorflow 2)

2.2 Pytorch

The results are very similar

dpi	cpi	unemp	invest	bias
0.276535	0.144858	-0.150715	-0.033078	0.00735

and the epoch/loss

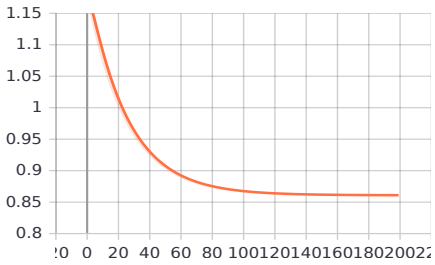

Figure 3: Epoch / Loss (Pytorch)