CS 4210

Classifying Income

USING A NEURAL NETWORK CONSIDERING CENSUS DATA

The Team

Ameer
Abdallah

Charles
Kypros

Paul
Vuong

Thuan
Lee

California Polytechnic University Pomona

CS 4210: Machine Learning and Its Applications

The Contents

01 IPUMS USA data set, catalogued by the ACS (American Community Surveys)

02 Machine Learning Model:
the *Neural Network*

03 How accurate was our model at classifying income?

04 Other related works in this field

Our data set

The data set was provided by IPUMS USA, a non-profit organization that produces democratized access to the world's social and economic data.

The data set was catalogued from 2019 data collected by the American Community Surveys.

IPUMS

AMERICAN COMMUNITY SURVEY
U.S. CENSUS BUREAU

The Neural Network

Machine Learning Model

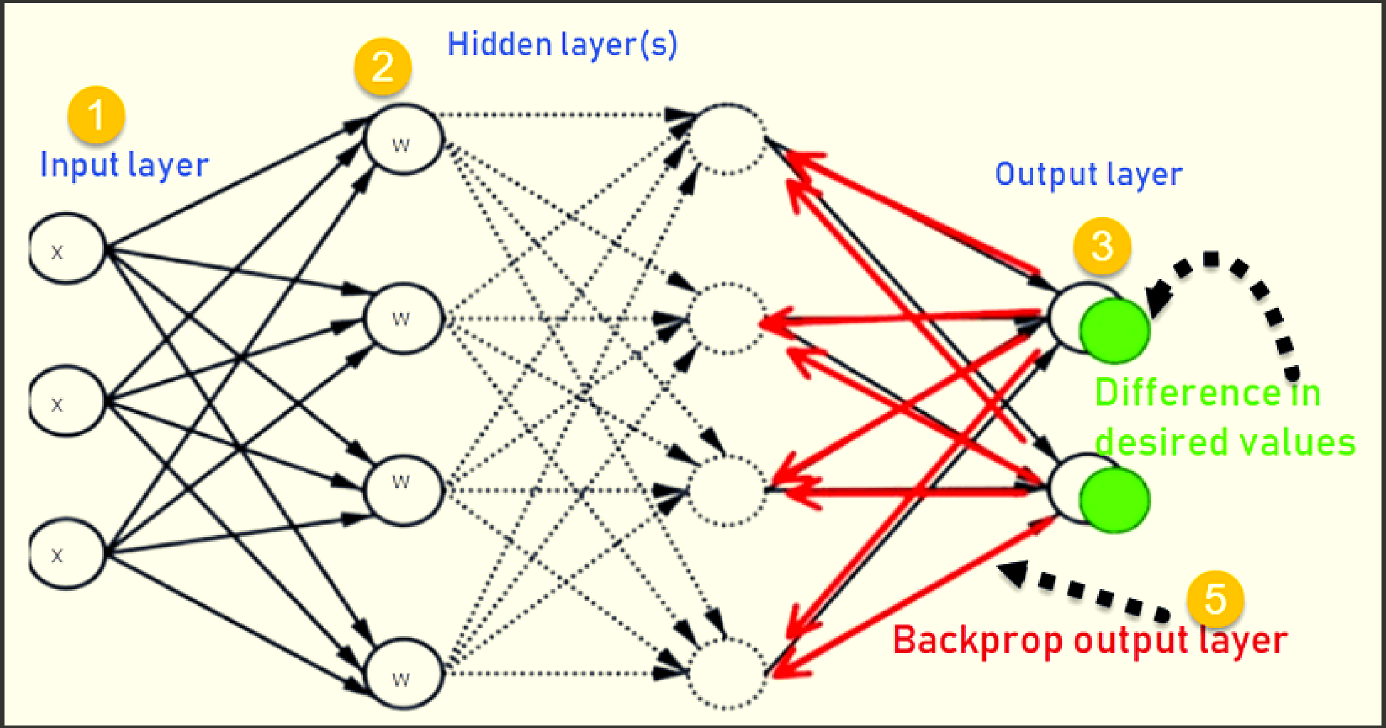
Start with random weights and biases to establish the learning model

Using training data, performance is evaluated to adjust weights and biases

Output nodes correspond to respective personal income brackets

Use stochastic gradient descent with Rectified Linear Unit activation function

Neural Network



We will classify only the personal income contribution. The amount of total layers and hidden layers that the neural network will contain is determined by testing and trials.

Our end classifier should be an income bracket prediction basedon multi-variable regression. We also introduced back-propagation to increase accuracy of the model.

Three Classifications

01

Under $48,500

02

$48,500 to $145,500

03

Over $145,500

Features

Sex

Age

Marital Status

Time Married

Race

Citizenship Status

Educational Attainment

Public or Private School

Degree Field

Employment Status

Class of Worker

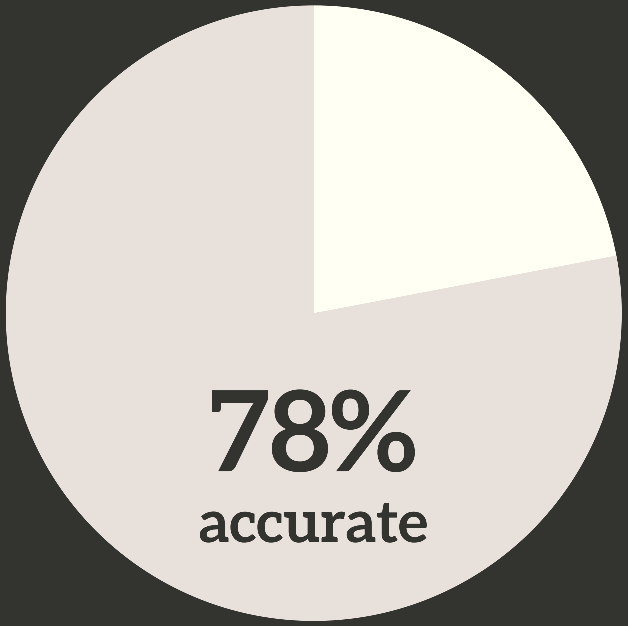
Hearing / Vision Difficulty

The results of our prediction model

EUR/USD - 1,35379 - 00:00:00 14 giu (EEST)
EUR/USD (Bid), Ticks, # 300 / 300
23:35
23:40
23:46
Gold, spot - 1.276,820 - 23:00:00 13 giu (CEST)
Gold, spot (Bid), 1 minute, # 159 / 300, Logarithmic, Heikin Ashi

- 13 unneccessary feature columns were removed.
- Set a maximum epoch of 1,000 which increases accuracy, but causes over fitting.
- Used a ratio of 67:33 ratio for training and testing respectively.
- Data set suffererd from missing values.

Resulting Accuracy



Our strongest result came at 78% accuracy when we used 10 hidden layers, 900 neurons, and 0.01 as our learning rate.

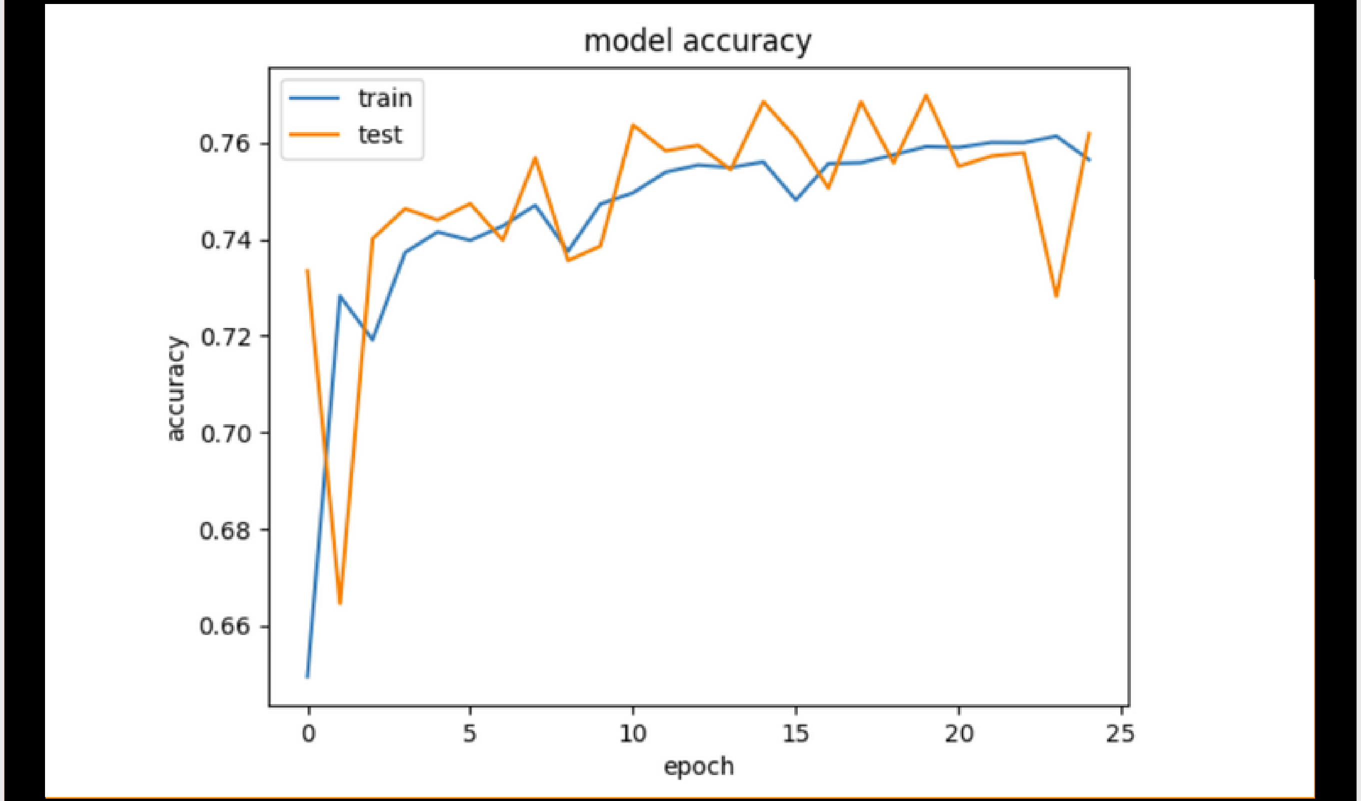


Figure: A comparison of model accuracy with epoch