Assignment 4: Report

ECON832 Computational Methods in Economics

Mar 29, 2024

Problem Set 4 involved using the `NN/ABKKExperiment_pilot.jl` script from the GitHub repo and implementing an alternative neural net model.

Task 1: Own model parameters

My model changed the initial parameters in the `NN/ABKKExperiment_pilot.jl` script in several ways. I changed the activation function to a *swish* function, which is a smooth approximation of the ReLU function. This swish function is defined as:

$$f(x) = x \cdot \sigma(x)$$

where $\sigma(x)$ is the sigmoid function. In the code, this is implemented within the `train()` function as:

```
swish(x,   = 1.0) = x * sigmoid( *x)
```

My loss function was changed from the logit entropy loss to logit binary cross entropy loss. This loss function is defined as:

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

where y is the true label and $\hat{y}$ is the predicted label. The main difference between the two loss functions is that the logit binary cross entropy loss function is more sensitive to the difference between the predicted and true labels. This means that the model will be penalized more for incorrect predictions. The logit binary cross entropy loss function is implemented across the code as follows.

```
using Flux: logitbinarycrossentropy ...
loss(x, y) = logitbinarycrossentropy(model(x), y)
```

The layers and width are kept the same since these depend on the data and problem at hand.

I changed the optimiser in the `train` function to the ADAM optimiser. This optimiser is a popular choice for training deep learning models and is known for its efficiency and speed. The ADAM optimiser is implemented in the code as follows.

```
optimiser = ADAM(0.001, (0.9, 0.8))
Flux.train!(loss, Flux.params(model), train_data, optimiser)
```

Task 2: Results

Below I show the results for my model training and test procedures.

Table 1: Confusion Matrix

Predicted class 1	**Predicted class 2**	**Predicted class 3**	**Predicted class 4**	**Predicted class 5**	**Predicted class 6**
999	9	21	0	6	3
3	0	138	250	101	0
207	151	97	54	86	287
7	40	124	322	89	2
0	157	8	8	112	197
0	133	16	7	138	327

Reported accuracies for the model are as follows

```
println("Training accuracy: ", in_sample_accuracy )
println("Test accuracy: ", in_test_accuracy)
```

```
Training accuracy: 0.45803854598682603
Test accuracy: 0.4530373261771164
```

Task 3: Comparison

Comparing to the QE paper, this machine learning approach can achieve a much larger accuracy in the prediction of choice than the linear mixture. However, the model is still not perfect and there is room for improvement. The model could be improved by attempting other new types of neural networks, such as convolutional neural networks or recurrent neural networks. These models could potentially capture more complex patterns in the data and improve the accuracy of the predictions. Additionally, the model could be improved by using more data and more features. This would allow the model to learn more about the data and make more accurate predictions. Overall, the machine learning approach is a promising method for predicting choice in the QE paper, but there is still room for improvement.

There are disadvantages to implementing machine learning in economic problems like these. Machine learning algorithms are "dumb" in the sense that they do not understand the underlying economic theory. This means that they may not be able to capture the complex relationships between variables that are present in economic models. Linear mixture models, such as the QE paper one, are more interpretable since they are built from underlying economi theory: in the paper we build models from RUM micro theory. Additionally, machine learning algorithms can be difficult to interpret, which can make it hard to understand why the model is making certain predictions. This can be a problem in economic problems where it is important to understand the underlying mechanisms that are driving the data.

Task 4: Risk aversion

Risk aversion is something that cannot be directly estimated from the data using the machine learning approach, unlike the methods based in theory such as the QE paper. However, we can use the machine learning model to make predictions about risk aversion. For example, we could use the model to predict the choice of an individual given different levels of risk aversion. By varying the level of risk aversion and observing how the predicted choice changes, we can infer the risk aversion of the individual. This is a more indirect way of estimating risk aversion than the methods based in theory, but it can still provide useful information about the individual's preferences.

Further, there are certain features that could be used which are produced by the model that could be used to estimate risk aversion. For example, the model could produce a feature that represents the uncertainty of the prediction. This feature could be used as a proxy for risk aversion, since individuals who are more risk averse are likely to be more uncertain about their choices. By using these features, we can estimate risk aversion using the machine learning model.