Homework I, Computational Macro

Be sure to include all the material that shows your work and to present your results in a way that is easy to understand. Remember: ignorance and inability to communicate are observationally equivalent.

1. Approximating a simple function (2 points)

Take the function $2x + 1.5x^2 - 2x^3$ defined in $[0, 2]$.

1. Approximate the function using a neural network with a hyperbolic activation function and 50 points randomly sampled in $[0, 2]$. Document the best selection of architecture in terms of the number of layers and neurons per layer.

2. Approximate the function using a neural network with a ReLU activation function and 50 points randomly sampled in $[0, 2]$. Document the best selection of architecture in terms of the number of layers and neurons per layer.

3. Compare the results in Steps 1. and 2. Do you have an intuition?

4. Repeat Steps 1. and 2. but now with 10,000 randomly selected grid points. Compare the results in Steps 1. and 2. as you change the size of the batch from SGD to pure GD.

2. Approximating a harder function (2 points)

Take the function $2x + 1.5x^2 - 2x^3$ if $x \in [0, 1]$ and $1 + 2x + 1.5x.^2 - 2x^3$ if $x \in [1, 2]$. Notice that now we have a discontinuity at 1.

Repeat Steps 1. to 4. from the previous exercise. Focus, when documenting your results, on the role of the discontinuity in the approximation.

3. Discrete choices (3 points)

A household i decides between vacationing at the sea (case 0) or the mountains (case 1) given a binary choice model:

$$choice = \begin{cases} 0 & \text{if } 0.3 * income_i + 0.2 * (age_i - 25) - 0.5 * size_i + \epsilon < 0 \\ 1 & \text{otherwise} \end{cases}$$

where $income_i$ is the income of the household, age_i is the age of the household head, and $size_i$ is the number of individuals in the household. Also, $\epsilon \sim \mathcal{N}(0, 0.1)$. In other words, richer, older, and smaller households tend to vacation by the sea, and poorer, younger, and larger households by the sea, but there is always a stochastic component.

Draw 500 households where $income_i$ is uniformly distributed between 0 and 20, age_i is uniformly distributed between 25 and 50, and $size_i$ is discretely uniformly distributed between 0 and 5.

1. Train a neural network to forecast the choices of households. Notice that you need to use a softmax layer in some moment.
2. Discuss your architecture choice.
3. Draw 200 new households with the same characteristics as above. Use the neural network to forecast their behavior. How good is your test error?
4. Robustness. Draw 200 new households, but now using the function:

$$choice = \begin{cases} 0 & \text{if } 0.2 * income_i + 0.3 * (age_i - 25) - 0.5 * size_i + \epsilon < 0 \\ 1 & \text{otherwise,} \end{cases}$$

   that is, income now is less important than before in the vacation choice, but age is more important. Use the same network and the *same* weights that you used above to forecast the behavior of these 200 new households. How well do you do? Is your original neural network robust?
5. Transfer learning. Take the weights from the network you trained in Step 1., and use them as initial values for training a new network with the 200 new households from Step 4. How fast is the convergence to the new network? Compare it in terms of speed to the network from Step 1.

4. Double descent (3 points)

Sample 50 points from the linear random function $y = 1 + x + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 0.1)$

1. Run a linear regression $y = a_0 + a_1 x$ on the simulated data. Which point estimates do you get?
2. Estimate a quadratic polynomial $y = a_0 + a_1 x + a_2 x^2$ and the simulated data. Which point estimates do you get?
3. Train a neural network with one layer and five neurons on the simulated data. Plot the resulting approximated function on a graph together with the 50 random observations and the linear and quadratic regressions.
4. Increase the number of layers and neurons of the neural network and retrain the network with the same 50 original random points. Plot the resulting approximated functions on a graph together with the 50 random observations and the linear and quadratic regressions. You can, for example, plot the new approximation each time you have 20 new weights in the network. Plot also the evolution of the train error.
5. Go as far as you can. For example, what is the form of the function approximated with a neural network with ten layers, each with ten neurons?