Econ 425; Week 8

Manu Navjeevan

February 23, 2021

1 Final Project

For the final project you will be using Machine-Learning to answer some question/make some prediction using data. Examples include using past financial history to predict credit card default or using economic indicators to predict recessions. In addition to coming up with a machine-learning model that does well at this task you will need to attach a write-up. This writeup should include:

- An abstract that gives a summary of the prediction task at hand and the approach taken

- An introduction that lays out the task at hand and explains why the task is important

- A summary of prior literature on this task (try to emphasize your contribution)

- A in depth discussion of the challenges you faced in completing the task and how you solved/dealt with them

- Analysis of your model and it's performance

- Conclusion and areas of future research

2 Recurrent Neural Networks

A recurrent neural network is a particular neural network structure that help us model data in some sort of sequence. Examples of this include:

- Time Series: here our data is a sequence of X_t and Y_t observations over time. Examples include stock prices or economic data

- Geographic Data: here our data is a sequence of X_t and Y_t observations over space. Examples may include unemployment rates in cities or the weather in a different areas

- Words in a text message, people's actions once arriving at a website, etc.

Importantly, the realization of Y at each step depends not only at the X realization at the step but also at the value of X at a previous step.

- Stock prices tommorow may not only depend on economic indicators tommorow, but also on economic indicators today.

- The unemployment rate in Burbank may not only depend on the economic conditions in Burbank itself but also on the economic conditions in LA.

When we come up with a model that relates X and Y in order to predict Y, it may be important to take this dependence structure into account.

A recurrent neural network takes into account the dependence structure across the sequence (across time-periods or geographies) by equipping the standard neural network that we've covered for the last few weeks with a *sequential memory*. We implement this by equipping the neural network with a "hidden state" or a

"memory" that is updated at step or each time period. This hidden state is used along with the characteristics X_t to make a prediction about Y_t, $\hat{Y}_t$

Essentially, at each step t we can thing of a recurrent neural network as doing two things:

1. Using hidden state and the characteristics X_t to make a prediction about Y_t

2. Updating the hidden state so that it can be used at the next step $(t+1)$

This is useful because it allows us to keep making predictions from one step to the next and take into account the dependence structure. We do not have to specify how long our sequence is, we can simply always use the hidden state along with the next observed characterisitcs X_t to make a prediction about the next outcome Y_t.
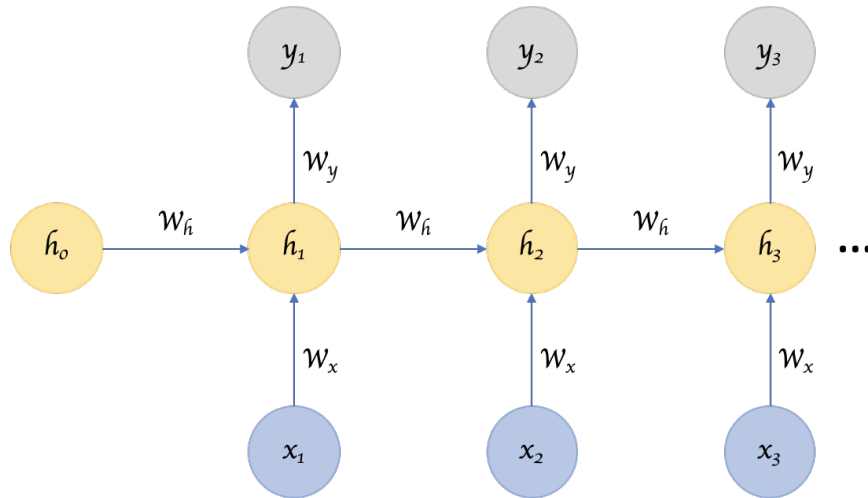

Figure 1: Illustration of the Graphical Structure of an RNN

So, when we are training a recurrent neural network we have to think about:

1. How to optimally use the hidden state and the characteristics to make the prediction (choose the weights of the feed forward neural network)

2. How to optimally update the hidden state

The first step we are familiar with how to do. However the second step is new and requires some more complicated gradient descent. One simple way around this is to specify how to update the hidden step. A simple approach would be to just specify that the hidden state contains the output from the previous step, $\hat{Y}_{t-1}$. This is the approach we will take in our empirical exercise.