Notes for the presentation

Parley Ruogu Yang

28 Jan 2021

1 What kind of AI can be easily tricked?

In general, we work with a continuous function $f : \mathbb{R}^d \to \mathbb{R}$, and adapt the usual metrics on $\mathbb{R}^d$, e.g. Euclidean metrics. We set up some mathematical and NN definitions as below.

Write $B(x; \varepsilon)$ as the open ball centred at x with radius ε. Write $\mathcal{NN}_{N,L}$ as the set of all L-layer neural networks with $N = (N_0, ..., N_L)$ number of neurons, with $N_0 = d$. A typical member $\phi \in \mathcal{NN}_{N,L}$ can be expressed as $\phi : \mathbb{R}^d \to \mathbb{R}$ where

$$\phi(x) = W_L \circ \rho \circ W_{L-1} \circ ... \circ \rho \circ W_1(x)$$

where W_l is affine for all l, and ρ is the activation function.

The training of the NN can be described as follows. Suppose we have access to a training set $X^{\texttt{train}} \subset \mathbb{R}^d$ and let there be a non-negative cost function $C(\cdot)$. Then the training aims to find a set of minimisers

$$\mathcal{M} := \underset{\phi \in \mathcal{NN}_{N,L}}{\arg\min}\, C(\{\phi(x), f(x)\}_{x \in X^{\texttt{train}}}) \tag{1}$$

Having introduced the general settings, we present a result below, which is an extension from Bastounis et al. (2021).

We work with a function $f : \mathbb{R}^d \to \mathbb{R}$ with its image covering $[0, 1]$, and assume the cost function to give 0 if and only if $\phi(x) = f(x)$ on all $x \in X^{\texttt{train}}$. The other technical assumptions on f and ρ are omitted here.

Theorem 1. *For any $\varepsilon > 0$ and any architecture $\mathcal{NN}_{N,L}$ with $L \geq 2$, there exists uncountably many non-intersecting training and testing data $X^{train}, X^{test} \subset \mathbb{R}^d$ such that*

- *Accuracy: there exists $\tilde{\phi} \in \mathcal{M}$ such that $\tilde{\phi}(x) = f(x)$ for all $x \in X^{train} \cup X^{test}$*
- *Instability: for all $\phi \in \mathcal{NN}_{N,L}$, there exists $x \in X^{train} \cup X^{test}$ such that there exists $v \in B(x; \varepsilon)$ such that $|\phi(v) - f(v)| \geq \frac{1}{2}$*

The proof contains mostly functional-analytic methods to trick the NN architecture. As a comment on the theorem, we read from the accuracy part as follows. Coherently with the empirical practices, we have an 'intelligent' by having a well-trained $\tilde{\phi}$ which predicts all results correctly from both the training and the test sets. However, the concerning point is the instability — one may easily twist a data point x to v such that the result damages largely. In fact, the twist can be arbitrarily small as $\varepsilon > 0$ is proposed arbitrarily.

This so far demonstrates the peril of the current architecture of NN, that one could comfortably observe the accuracy of certain training methods, accompanied by the instability potentials where predictions go terribly wrong. The theorem above requires some technical assumptions on f, and a key question lies at how to best generalise cases where the theorem holds and where it does not. Mathematically, given a connected subset $D \subset \mathbb{R}^d$, I aim to partition, say all continuous functions $C(D;\mathbb{R})$ by sets $\mathcal{F}^{\texttt{stable}}$ and $\mathcal{F}^{\texttt{unstable}}$. This would offer more insights on what kind of functions could be stably approximated and the others that could not.

2 Can we have stable AI for forecasting purposes?

Forecasting is a crucial topic in finance — one obtains predictions about the future state given the current information. This could be, say in the financial market, a price forecast on certain asset at 11AM based on the historical price and other instruments until 10AM; in the context of macroeconomics, this could be forecasting the economic growth for the upcoming quarter based on all socio-economic data available today. Such a task can be mathematically summarised as

$$y_{t+k|t} = f(\Phi_t; \theta_t; h_t) \tag{2}$$

The left hand side is the variable of interest — a prediction of the future state y_{t+k} based on all information up to time t, whereas on the right, f maps three key components into the forecast: information (Φ_t), parameters (θ_t), and functional form (h_t).

Example: A discrete AR(1) model

$$y_t = \mu + \rho y_{t-1} + \varepsilon_t$$

where ε_t is a zero mean σ^2 variance time series process. If this were the true Data Generating Process (DGP), then we have $y_{t+1|t} = \mu + \rho y_t$. Under the framework of Equation 2, we have $\Phi_t = \{y_\tau : \tau \leq t\}$ and $\theta_t = (\mu, \rho, \sigma)$ with h_t enforcing the linear combination so that $f(\Phi_t; \theta_t; h_t) = \mu + \rho y_t$.

Certainly, in more sophisticated NN architecture, we have a set of equations with potential hidden neurons, but all of which can be summarised as a function

$f(\cdot;\cdot;h_t)$ as h_t specifies the functional form. With a given functional form, we train the model using some or all of the information available (Φ_t). In traditional models, we may simply optimise the maximum likelihood of the chosen dataset and obtain a Maximum Likelihood Estimator, and in more modern AI models, we use cost-minimisation techniques to find the best parameter θ_t. These can be represented by Equation 1.

Now, it is clear that the instability issue faced in Theorem 1 is inherited in time series AI models. In fact, there is one more layer of instability, which is the issue of stationarity, e.g. whether y_t has a finite expectation. In an AR(1) model, this means $|\rho| < 1$ and depending on which parametric specification is chosen, we have different restrictions on the parameters.

Hence, in the context of time series, there can be an additional statement being added on top of the two in Theorem 1, which is non-stationarity. One way to state that is as follows. Suppose the entire time series $\{y_t\}_{t\in\mathbb{N}}$ is stationary. Now, under certain functional form $f(\cdot;\cdot;h_t)$, we may produce a sequence of forecasts $\{y_{t+k|t}\}_{k=1}^{K}$ such that $\exists k' \in \{1, ..., K-1\}$ with $y_{t+k|t}$ being non-stationary for all $k > k'$.

For issues like this, one potential solution is to filter out the functional classes that appear to be unstable based on the previous information (Yang 2020). For instance, if the true DGP were ARIMA(1,1,0), then an ARIMA(0,0,1) model can be thought as unstable by the nature of the function. The search of a good function h_t can then be restricted to a 'reliable' set, constructed by excluding the unstable ones.

Computing procedures like that could help to reduce the non-stationarity. Nonetheless, a general method based upon a more precise and theoretically conceivable theory is still desired. What is also massively undiscovered is the treatment towards more sophisticated AI models where instability and non-stationarity coexist. This should be sourced from the findings in the first topic of research — a full categorisation of $\mathcal{F}^{\texttt{stable}}$ and $\mathcal{F}^{\texttt{unstable}}$, which can then be extended into the world of parametric restrictions that relates more closely to the issue of stationarity in time series.

3 What can we learn from the adversarial attacks?

The defend, that is, the response to an adversarial attack on time series AI can be forged upon rigorous results from the previous two pieces of research. Say we are interested in obtaining a forecast $y_{T+k|T} \in \mathbb{R}$ based on many potential time series variables. In particular, we may consider a computing procedure as outlined below:

1. Write $\mathcal{T}$ as the set of all back-testing points, identify the common domain of the potential function. That is,

$$g : D \to \mathbb{R} \text{ so that } \forall t \in \mathcal{T}, y_{t+k|t} = g(x_t)$$

2. Partition and categorise the functional space $C(D;\mathbb{R})$ into $\mathcal{F}^{\texttt{unstable}}$ and $\mathcal{F}^{\texttt{stable}}$

3. For all $t \in \mathcal{T}$, produce $\mathcal{F}_t^{\texttt{stable}}$ by the following procedure:

   (a) For each $f(\cdot;\cdot;h_t) \in \mathcal{F}^{\texttt{stable}}$, fit the data and produce $\theta_t(h_t)$ with forecasts.

   (b) Judge if the fit and forecasts under each $f(\cdot;\cdot;h_t) \in \mathcal{F}^{\texttt{stable}}$ is stationary. If not, deduce it from the set $\mathcal{F}^{\texttt{stable}}$.

   (c) Therefore obtain a subset $\mathcal{F}_t^{\texttt{stable}} \subset \mathcal{F}^{\texttt{stable}}$.

4. Obtain $H := \bigcap\{\mathcal{F}_t^{\texttt{stable}} : t \in \mathcal{T}\}$, then for all $h_T \in H$, find the optimised parameter $\theta_T(h_T)$ and obtain the best candidate $y_{T+k|T}$.

We see that the item 2 to heavily rely on the categorisation method to be investigated in the first topic, and the item 3 depends on the nature of time series stationarity, which relates to the second topic. In particular, the implementation of 3(b) could be via hypothesis testing, where we have the null hypothesis as $f(\cdot;\cdot;h_t) \in \mathcal{F}_t^{\texttt{stable}}$ and derive the theoretical distribution of the fit and forecast under the null to reach a conclusion of either rejection or acceptance.

References

Bastounis, A., Anders C. Hansen, and V. Vlacic (2021). "On Computational barriers and paradoxes in estimation, regularisation, learning and computer assisted proofs." In: Work In Progress.

Yang, Parley Ruogu (2020). "Using the yield curve to forecast economic growth." In: *Journal of Forecasting* 39.7, pp. 1057–1080. DOI: 10.1002/for.2676.