# Prediction Inference of Time Series with Standard ReLU Deep Neural Networks

Kejin Wu

Department of Mathematics and Statistics
Loyola University Chicago

August 11, 2026

# Prediction of time series

- **Time series** is a discrete-time stochastic process, i.e., $\{X_t, t \in \mathbb{Z}\}$. Its realization is called time series data, e.g., heights of ocean tides, and counts of sunspots.
- **Prediction inference** is about determining the Optimal Predictor (OP), usually in $L_2$ or $L_1$ sense, and the Prediction Interval (PI), percentile or centered version, of future value $X_{n+j}$, $j \geq 1$, based on observed $\{X_1, \ldots, X_n\}$. We consider the Coverage Rate (CVR) and Length (LEN) of PI.

# Example

**Time series model**:

We assume that the time series data is generated by some underlying mechanism:

$$X_t = G(\boldsymbol{X}_{t-d}, \epsilon_t), \tag{1}$$

where:

- $G(\cdot, \cdot)$ could be any suitable linear/non-linear function that makes the time series well-behaved.
- $\epsilon_t$ is called innovation and assumed to be *i.i.d.* with appropriate moments and independent with $X_{t-i}$, $i \geq 1$.
- $\boldsymbol{X}_{t-d}$ represents $\{X_{t-1}, \ldots, X_{t-d}\}$.

# Example

**Linear models**, e.g., $X_t = \sum_{i=1}^{d} a_i X_{t-i} + \epsilon_t$; $\epsilon_t \overset{i.i.d.}{\sim} N(0, \sigma_\epsilon^2)$.

- One-step-ahead $L_2$ conditional OP $X_{n+1}^{L_2}$ is:

$$\mathbb{E}(X_{n+1}|X_n, \ldots, X_1) = \sum_{i=1}^{d} a_i X_{n+1-i}.$$

  Iterating this prediction can get multi-step-ahead $L_2$ OP $X_{n+j}^{L_2}$.

- A $(1-\alpha)\cdot 100\%$ PI of $X_{n+j}$ centered at $X_{n+j}^{L_2}$ can be expressed as:

$$\left[X_{n+j}^{L_2} - z_{\alpha/2}\sqrt{\text{Var}[e_n(j)]}, X_{n+j}^{L_2} + z_{\alpha/2}\sqrt{\text{Var}[e_n(j)]}\right],$$

  where $e_n(j)$ is $X_{n+j} - X_{n+j}^{L_2}$.

**Limitations:** Infeasible to incorporate non-linear models; rely on the normality assumption; hard to figure out the multi-step-ahead OP $X_{n+1}^{L_1}$.

# Example

**Non-linear models**, e.g., $X_t = m(\boldsymbol{X}_{t-d}) + \epsilon_t$.

- One-step-ahead $L_2$ OP $X_{n+1}^{L_2} = m(\boldsymbol{X}_{n+1-d})$.
- For multi-step ahead prediction, the iterative method is no longer $L_2$ optimal. Pemberton (1987) should be the first attempt to get exact $L_2$ OP based on numerical integral. Guo et al. (1999) further developed an analytical predictor based on innovation distribution $F_\epsilon$ to approximate the exact $L_2$ OP.

**Limitations:** Hard to find $L_1$ OP and PI.

# Oracle prediction with simulation

Take general model (1) as an example:

- Simulate $\{\epsilon_{n+1}^{(i)}, \ldots, \epsilon_{n+J}^{(i)}\}_{i=1}^{M}$ from $F_\epsilon$.
- Compute pseudo $\{X_{n+j}^{(i)}\}_{i=1}^{M}$, i.e., $X_{n+j}^{(i)} = G(\boldsymbol{X}_{T+j-d}, \epsilon_{n+j}^{(i)})$, for $j = 1, \ldots, J$.
- Take sample mean and median of $\{X_{n+j}^{(i)}\}_{i=1}^{M}$ to approximate $X_{n+j}^{L_2}$ and $X_{n+j}^{L_1}$, respectively; take corresponding quantile values to approximate PIs with arbitrary coverage rates.

**Limitations**: In practice, model information is generally not known to participants. Thus, this prediction is *Oracle*.

# Practical prediction with bootstrap

Assume we can get consistent estimators $\widehat{G}(\cdot,\cdot)$ and $\widehat{F}_\epsilon$ which is the empirical distribution of residuals, then: Apply Bootstrap to do predictions:

- Bootstrap $\{\hat{\epsilon}_{n+1}^{(i)},\ldots,\hat{\epsilon}_{n+j}^{(i)}\}_{i=1}^{M}$ from $\widehat{F}_\epsilon$.
- Compute pseudo $\{\widehat{X}_{n+j}^{(i)}\}_{i=1}^{M}$ iteratively, i.e., $\widehat{X}_{n+j}^{(i)} = \widehat{G}(\boldsymbol{X}_{T+j-d}, \hat{\epsilon}_{n+j}^{(i)})$, for $j = 1,\ldots,J$.
- Take sample mean and median of $\{\widehat{X}_{n+j}^{(i)}\}_{i=1}^{M}$ to approximate $X_{n+j}^{L_2}$ and $X_{n+j}^{L_1}$, respectively; take corresponding quantile values to approximate PIs with arbitrary coverage rates; we call it Quantile PI (QPI).

**Limitations**: (1) With finite samples, this Bootstrap-based PI suffers from the undercoverage issue; (2) the difficulty to get consistent estimations on the model and distribution of innovation.

# Forward bootstrap prediction for the general model

Motivated by Politis (2015), we propose a bootstrap prediction method for the general model (1), i.e., $X_n = G(\boldsymbol{X}_{t-d}, \epsilon_n)$.

**Main Algorithm**:

- *Step 1*: Do estimations to get $\widehat{G}(\cdot,\cdot)$ and $\widehat{F}_\epsilon$. Then, perform the bootstrap prediction to get $\widehat{X}_{n+j}$ for $j = 1, \ldots, J$.
- *Step 2*: Generate a whole pseudo series $\{X_1^*, \ldots, X_{n+J}^*\}$ by viewing $\widehat{G}(\cdot,\cdot)$ and $\widehat{F}_\epsilon$ as the true model and innovation distribution in the bootstrap world.
- *Step 3*: Re-estimate model to get $\widehat{G}^*(\cdot,\cdot)$ with $\{X_1^*, \ldots, X_n^*\}$; Re-define $\{X_{n-d+1}^* = X_{n-d+1}, \ldots, X_n^* = X_n\}$. Then do the bootstrap prediction with $\widehat{G}^*(\cdot,\cdot)$ and $\widehat{F}_\epsilon$ to get $\widehat{X}_{n+j}^*$. Record the predictive root $X_{n+j}^* - \widehat{X}_{n+j}^*$ in the bootstrap world.
- *Step 4*: Repeat Steps 2 and 3 $M$ times, collect $M$ predictive roots and take its empirical distribution to approximate the distribution of $X_{n+j} - \widehat{X}_{n+j}$.
- *Step 5*: The $(1-\alpha)100\%$ **Pertinent PI** for $X_{n+j}$ centered at $\widehat{X}_{n+j}$ can be approximated by $[\widehat{X}_{n+j} + q(\alpha/2), \widehat{X}_{n+j} + q(1-\alpha/2)]$, where $q(\alpha)$ is the $\alpha$-quantile of the empirical distribution of $X_{n+j}^* - \widehat{X}_{n+j}^*$.

# Forward bootstrap prediction for additive model

We consider a variant of model (1):

$$X_t = G\left(\boldsymbol{X}_{t-d}, \epsilon_t\right) := f_0(\boldsymbol{X}_{t-d}) + \epsilon_t.$$

Typically, $f_0(\cdot)$ can be estimated by the standard kernel estimator.

We focus on the estimator as the fully connected Multi-layer Perceptron, i.e., one type of Deep Neural Networks (DNN).

**Remark:**

- Deep or shallow neural networks possess the universal approximation property; see Cybenko (1989); Yarotsky (2018).
- DNN can better conquer the curse of dimensionality than standard kernel methods; see Bauer and Kohler (2019); Kohler et al. (2019).

# The structure of Deep Neural Networks

We can write a DNN function $f_{\text{DNN}}$ as

$$f_{\text{DNN}}(\boldsymbol{x}) = \boldsymbol{A}_{L+1}(\phi(\boldsymbol{A}_L(\cdots\phi(\boldsymbol{A}_3\phi(\boldsymbol{A}_2\phi(\boldsymbol{A}_1\boldsymbol{x}+\boldsymbol{b}_1)+\boldsymbol{b}_2)+\boldsymbol{b}_3)\cdots)+\boldsymbol{b}_L)+\boldsymbol{b}_{L+1};$$

where $\{\boldsymbol{A}_i\}_{i=1}^{L+1}$ are weight matrices whose shapes depend on the number of hidden neurons and output dimension; $\{\boldsymbol{b}_i\}_{i=1}^{L+1}$ are intercept terms; $\phi(\cdot)$ is the activation function.

The class of DNN we consider is denoted by:

$$\begin{aligned}\mathcal{N}_n &= \mathcal{N}(L_n, H_n, 2B_n)\\ &:= \{f_{\text{DNN}}(\boldsymbol{x}) : \text{ with depth at most } L_n \text{ and equal width at most } H_n; \|f_{\text{DNN}}(\boldsymbol{x})\|_\infty \le 2B_n\};\end{aligned}$$

# Assumptions

- A1. The target function $f_0$ lies in the Hölder space $C^{k,\alpha}\left([-B_n, B_n]^d\right)$, which is the space of $k$ times continuously differentiable functions on $[-B_n, B_n]^d$ having a finite Hölder norm defined by $\|f\|_{C^{k,\alpha}([-B_n,B_n]^d)}$. In particular, we define the function space $\mathcal{F}_{r,d,[-B_n,B_n]} := \{f : \|f\|_{C^{k,\alpha}([-B_n,B_n]^d)} < \infty \text{ and } \|f\|_\infty \leq B_n\}$, where $r = k + \alpha$; $f_0 \in \mathcal{F}_{r,d,[-B_n,B_n]}$.
- A2. The whole series $\{X_t\}$ is strictly stationary and geometrically $\beta$-mixing. We denote the stationary distribution of $\{X_t\}$ by $\mathbb{P}_\pi$ and we assume it is absolutely continuous.
- A3. $\lim_{n\to\infty} \log(n) \cdot \mathbb{P}_\pi \left(\max_{t\in\{1,\ldots,n\}} |Y_t| \geq B_n\right) = 0$.

# Oracle estimation ability of DNN

**Lemma 1:** Universal approximation property of DNN

Under assumption A1, when $n$ is sufficiently large, we have

$$\forall f_0 \in \mathcal{F}_{r,[-B_n,B_n]^d},\ \exists f^*_{DNN} \in \mathcal{N}(L_n, H_n, (2B_n)^{r+1}),\ s.t.,\ \|f^*_{DNN} - f_0\|_\infty \leq (2B_n)^r n^{-\eta\frac{r}{d}};$$

where $L_n = O(\log n)$; $H_n = O(n^\eta \log n)$ and $\eta \in (0,1)$ is an appropriate constant.

The rate of $\infty$-norm error converges to 0, depending on the behavior of the sequence $(2B_n)^r n^{-\eta\frac{r}{d}}$ as $n \to \infty$.

# $L_2$ convergence of DNN

**Theorem 1:** $L_2$ error bound of DNN

Under assumptions A1 to A3, taking $B_n \asymp n^{K_B}$ for $K_B \in [0, \frac{1}{2(r+d)+2})$, and setting $L_n \asymp \log(n)$ and $H_n \asymp n^{(\frac{d}{r+d})(1/2-K_B)} \log(n)$. Then, if the DNN estimator satisfies $Q_n\left(\hat{f}_n\right) \leq \inf_{f \in \mathcal{F}_n} Q_n(f) + \theta_n$; where $Q_n(\cdot) := \frac{1}{n}\sum_{t=1}^{n}(X_n - f(\boldsymbol{X}_{t-1}))^2$, and $\epsilon_n = C\left(n^{rK_B - \frac{r}{r+d}(1/2-K_B)} \log^4(n) + \sqrt{\mu_n + \theta_n}\right)$; $\mu_n \to 0$ and $C > 0$ is a constant independent of $n$. Then, for sufficiently large $n$, we have

$$\left\|\hat{f}_n - f_0\right\|_{\mathcal{L}^2} \leq \epsilon_n,$$

with probability at least $\xi_n$; $\xi_n \to 1$ as $n \to \infty$.

# From $L_2$ convergence to pointwise convergence

We attempt to make PI conditional on the latest observations. Thus, the $L_2$ error bound, which quantifies the average performance of an estimator over $\boldsymbol{X}$, is not sufficient to give pointwise estimation inference.

**Additional assumptions:**

- C1. There exist $\mathcal{L}_n < \infty$ and events $\mathcal{E}_n$ with $P(\mathcal{E}_n) \to 1$ such that on $\mathcal{E}_n$,

$$\left|\hat{f}_n(\boldsymbol{x}) - \hat{f}_n(\boldsymbol{x}')\right| \leq \mathcal{L}_n \left\|\boldsymbol{x} - \boldsymbol{x}'\right\| \quad \forall \boldsymbol{x}, \boldsymbol{x}' \in [-B_n, B_n]^d.$$

- C2. The joint density of $\boldsymbol{X}_{t-1}$ has a positive lower bound, i.e., $\pi(\boldsymbol{x}_{t-1}) \geq C_\pi > 0$ for all $\boldsymbol{x}_{t-1}$ in its domain, where $C_\pi$ is a constant.
- C3. $\epsilon_n$ defined in Theorem 1 and the Lipschitz constant of $g := \hat{f} - f_0$, namely $\tilde{\mathcal{L}}_n$ satisfy $\frac{(\tilde{\mathcal{L}}_n)^d \epsilon_n^2}{C_\pi} \longrightarrow 0$ as $r \geq 1$ and $n \to \infty$.

# Pointwise estimation inference of DNN

**Lemma 2:** Pointwise Estimation inference of DNN

With all conditions assumed in Theorem 1 and C1 to C3, we have

$$\hat{f}_n(\boldsymbol{x}) \xrightarrow{p} f_0(\boldsymbol{x}), \text{ as } n \to \infty$$

for $\forall\ \boldsymbol{x} \in [-B_n, B_n]^d$.

# Consistent point prediction and asymptotically valid QPI

With further assumptions:

- A4. The probability density of innovation $\epsilon_t$, namely $p_\epsilon(x)$, is continuously differentiable and satisfies $\sup_x p_\epsilon(x) < \infty$, and $|p_\epsilon(a) - p_\epsilon(b)| \leq L|a-b|$ for a finite constant $L$ and any $a$ and $b$ belong to the domain.
- A5. $F_{X_{n+j}|X_n,\ldots,X_1}(x)$ is continuous and strictly increasing w.r.t. any $x$ for each $j$.
- A6. $B_n$ increases in an appropriate rate s.t., A3 is satisfied and $B_n \cdot \sup_{|x|\leq B_n} |\widehat{F}_{X^*_{n+j}|X_n,\ldots,X_1}(x) - F_{X_{n+j}|X_n,\ldots,X_1}(x)| = op(1)$.

**Lemma 3:** Consistent point prediction and asymptotically valid QPI

With all conditions assumed in Lemma 2, A4 A5 and A6, $L_2$ and any quantile point predictions are consistent. Meanwhile, the QPI is asymptotically valid.

# Preparation for PPI

To get the PPI, we need to show:

(1) The pseudo series $\{X_1^*, \ldots, X_{T+k}^*\}$ generated by $\hat{f}_n$ and $\widehat{F}_\epsilon$ is still geometrically $\beta$-mixing and its stationary distribution converges to the original one in probability.

(2) Denote $A_{n+j} = f_0(\boldsymbol{Y}_{n+j-1}) - \hat{f}_n(\boldsymbol{Y}_{n+j-1})$ which represents the $j$-step ahead estimation error. The distribution of $a_n A_{n+j}$ converges to a non-trivial continuous distribution for some appropriate sequence $a_n \to \infty$ as $n \to \infty$.

- (1) can be shown in the spirit of Franke et al. (2002, 2004), which claims that the model-based bootstrap procedure can mimic the complete dependence structure of the original time series.
- (2) stands for a *minimal* assumption to derive the PPI. This kind of assumption is common in the subsampling estimation area, where this assumed distribution is proposed to be approximated by subsampling; see Politis et al. (1999); Wolf and Wunderli (2015); Politis (2024).

# Simulations

We consider the three simulation models as follows:

- Model 1: $Y_n = \log(Y_{t-1}^2 + 1) + \epsilon_t$;
- Model 2: $Y_n = \sin(\sum_{i=1}^{5} Y_{t-i}) + \epsilon_t$;
- Model 3: $Y_n = \sin(\sum_{i=1}^{10} Y_{t-i}) + \epsilon_t$;

We build QPI and PPI with DNN and standard kernel estimators.

We take `optim.Adam` from `torch` package to train DNN with 100 epochs, 10 batch size and 0.0002 learning rate; we take `KernelReg` function from `statsmodels` to make a kernel estimator and the bandwidth is chosen by the least-squares cross-validation method.

We consider sample size $n = 50, 200, 500$. When $n$ is small, "high-dimension" situation is mimicked.

We evaluate different PIs by the CoVerage Rate (CVR) and Length (LEN) for each step ahead prediction horizon, and Mean Squared Error (MSE) for $L_2$ point prediction:

$$\text{CVR}_j = \frac{1}{R}\sum_{i=1}^{R} \mathbb{1}(Y_{i,n+j} \in [\hat{L}_{i,j}, \hat{U}_{i,j}]);\ \text{LEN}_j = \frac{1}{R}\sum_{i=1}^{R}(\hat{U}_{i,j} - \hat{L}_{i,j});\ \text{MSE}_j = \frac{1}{R}\sum_{i=1}^{R}(Y_{i,j} - \hat{Y}_{i,j})^2;$$

where $R$ is the number of replications, which is taken as 1000.

We fix the significance level to be 0.05.

# Simulation results with Model 1

| Step | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | DNN | | | | | Kernel | | |
| Model-1; n = 50 | | | | | | | | | | |
| CVR-QPI | 0.927 | 0.898 | 0.901 | 0.884 | 0.889 | 0.886 | 0.906 | 0.923 | 0.902 | 0.906 |
| LEN-QPI | 4.105 | 4.352 | 4.353 | 4.355 | 4.354 | 3.533 | 4.385 | 4.552 | 4.607 | 4.635 |
| CVR-PPI | 0.941 | 0.902 | 0.904 | 0.893 | 0.897 | 0.915 | 0.913 | 0.921 | 0.910 | 0.913 |
| LEN-PPI | 4.276 | 4.376 | 4.377 | 4.376 | 4.381 | 4.068 | 4.552 | 4.669 | 4.717 | 4.739 |
| MSE | 1.259 | 1.604 | 1.640 | 1.795 | 1.751 | 1.203 | 1.573 | 1.593 | 1.794 | 1.739 |
| Model-1; n = 200 | | | | | | | | | | |
| CVR-QPI | 0.935 | 0.943 | 0.955 | 0.934 | 0.945 | 0.931 | 0.945 | 0.952 | 0.936 | 0.944 |
| LEN-QPI | 3.843 | 4.524 | 4.712 | 4.780 | 4.810 | 3.817 | 4.536 | 4.714 | 4.773 | 4.798 |
| CVR-PPI | 0.932 | 0.944 | 0.955 | 0.931 | 0.942 | 0.940 | 0.945 | 0.959 | 0.932 | 0.936 |
| LEN-PPI | 3.875 | 4.542 | 4.740 | 4.802 | 4.825 | 3.955 | 4.577 | 4.746 | 4.793 | 4.815 |
| MSE | 1.066 | 1.339 | 1.421 | 1.674 | 1.677 | 1.101 | 1.343 | 1.420 | 1.667 | 1.677 |
| Model-1; n = 500 | | | | | | | | | | |
| CVR-QPI | 0.950 | 0.940 | 0.944 | 0.941 | 0.954 | 0.956 | 0.945 | 0.941 | 0.937 | 0.953 |
| LEN-QPI | 3.875 | 4.594 | 4.776 | 4.840 | 4.864 | 3.870 | 4.585 | 4.767 | 4.828 | 4.847 |
| CVR-PPI | 0.953 | 0.938 | 0.946 | 0.940 | 0.952 | 0.955 | 0.943 | 0.941 | 0.935 | 0.952 |
| LEN-PPI | 3.882 | 4.603 | 4.786 | 4.841 | 4.866 | 3.935 | 4.600 | 4.768 | 4.825 | 4.842 |
| MSE | 0.942 | 1.474 | 1.545 | 1.555 | 1.556 | 0.964 | 1.460 | 1.522 | 1.535 | 1.536 |

Table 1: Simulation results with Model-1 data.

# Simulation results with Model 2

| Step | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | DNN | | | | | Kernel | | |
| Model-2; n = 50 | | | | | | | | | | |
| CVR-QPI | 0.907 | 0.920 | 0.938 | 0.929 | 0.931 | 0.546 | 0.595 | 0.681 | 0.724 | 0.754 |
| LEN-QPI | 4.417 | 4.489 | 4.602 | 4.680 | 4.707 | 2.453 | 2.841 | 3.167 | 3.431 | 3.635 |
| CVR-PPI | 0.920 | 0.922 | 0.943 | 0.931 | 0.938 | 0.693 | 0.712 | 0.750 | 0.766 | 0.802 |
| LEN-PPI | 4.634 | 4.655 | 4.703 | 4.764 | 4.800 | 4.176 | 4.202 | 4.243 | 4.345 | 4.358 |
| MSE | 1.639 | 1.615 | 1.498 | 1.657 | 1.542 | 2.352 | 2.180 | 1.985 | 2.133 | 1.888 |
| Model-2; n = 200 | | | | | | | | | | |
| CVR-QPI | 0.877 | 0.894 | 0.922 | 0.911 | 0.914 | 0.755 | 0.835 | 0.877 | 0.908 | 0.928 |
| LEN-QPI | 4.051 | 4.185 | 4.292 | 4.335 | 4.397 | 3.457 | 3.942 | 4.269 | 4.468 | 4.638 |
| CVR-PPI | 0.895 | 0.912 | 0.931 | 0.914 | 0.916 | 0.826 | 0.861 | 0.898 | 0.907 | 0.928 |
| LEN-PPI | 4.359 | 4.405 | 4.436 | 4.439 | 4.451 | 4.705 | 4.746 | 4.767 | 4.775 | 4.811 |
| MSE | 1.709 | 1.576 | 1.636 | 1.619 | 1.519 | 2.002 | 1.777 | 1.821 | 1.722 | 1.553 |
| Model-2; n = 500 | | | | | | | | | | |
| CVR-QPI | 0.880 | 0.879 | 0.899 | 0.916 | 0.931 | 0.801 | 0.873 | 0.913 | 0.927 | 0.936 |
| LEN-QPI | 3.747 | 4.067 | 4.244 | 4.354 | 4.431 | 3.391 | 4.032 | 4.440 | 4.617 | 4.747 |
| CVR-PPI | 0.910 | 0.906 | 0.911 | 0.922 | 0.931 | 0.844 | 0.891 | 0.912 | 0.926 | 0.937 |
| LEN-PPI | 4.176 | 4.375 | 4.451 | 4.493 | 4.520 | 4.639 | 4.766 | 4.857 | 4.835 | 4.862 |
| MSE | 1.494 | 1.629 | 1.600 | 1.564 | 1.503 | 1.732 | 1.717 | 1.680 | 1.598 | 1.520 |

Table 2: Simulation results with Model-2 data.

# Simulation results with Model 3

| Step | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | DNN | | | | | Kernel | | |
| Model-3; n = 50 | | | | | | | | | | |
| CVR-QPI | 0.897 | 0.912 | 0.918 | 0.894 | 0.933 | 0.021 | 0.018 | 0.022 | 0.035 | 0.034 |
| LEN-QPI | 4.343 | 4.407 | 4.441 | 4.483 | 4.534 | 0.078 | 0.104 | 0.116 | 0.146 | 0.169 |
| CVR-PPI | 0.918 | 0.920 | 0.934 | 0.899 | 0.936 | 0.150 | 0.143 | 0.157 | 0.183 | 0.163 |
| LEN-PPI | 4.600 | 4.601 | 4.589 | 4.603 | 4.643 | 1.160 | 1.211 | 1.208 | 1.257 | 1.330 |
| MSE | 1.641 | 1.655 | 1.479 | 1.706 | 1.494 | 3.004 | 2.919 | 2.896 | 2.957 | 2.839 |
| Model-3; n = 200 | | | | | | | | | | |
| CVR-QPI | 0.848 | 0.859 | 0.879 | 0.869 | 0.879 | 0.521 | 0.582 | 0.628 | 0.630 | 0.651 |
| LEN-QPI | 3.950 | 4.008 | 4.077 | 4.118 | 4.162 | 2.392 | 2.599 | 2.798 | 2.960 | 3.129 |
| CVR-PPI | 0.866 | 0.896 | 0.913 | 0.892 | 0.896 | 0.639 | 0.665 | 0.731 | 0.709 | 0.729 |
| LEN-PPI | 4.348 | 4.365 | 4.396 | 4.383 | 4.393 | 3.981 | 3.983 | 4.008 | 4.041 | 4.085 |
| MSE | 1.813 | 1.721 | 1.677 | 1.804 | 1.715 | 2.315 | 2.308 | 2.040 | 2.307 | 2.075 |
| Model-3; n = 500 | | | | | | | | | | |
| CVR-QPI | 0.961 | 0.933 | 0.934 | 0.943 | 0.954 | 0.869 | 0.880 | 0.896 | 0.902 | 0.942 |
| LEN-QPI | 4.633 | 4.641 | 4.650 | 4.656 | 4.667 | 4.054 | 4.277 | 4.407 | 4.482 | 4.579 |
| CVR-PPI | 0.955 | 0.931 | 0.933 | 0.937 | 0.945 | 0.898 | 0.881 | 0.889 | 0.902 | 0.933 |
| LEN-PPI | 4.645 | 4.651 | 4.652 | 4.660 | 4.649 | 5.020 | 4.972 | 4.894 | 4.851 | 4.835 |
| MSE | 1.359 | 1.533 | 1.641 | 1.472 | 1.438 | 1.686 | 1.752 | 1.819 | 1.683 | 1.545 |

Table 3: Simulation results with Model-3 data.

# Summary

- We can get consistent point predictions and asymptotically valid prediction intervals with the DNN estimator.
- Under the so-called “minimal assumption”, the pertinent prediction interval is possible with the DNN estimator.
- Predictions with the DNN estimator is more robust against the curse of dimensionality compared to applying the standard kernel estimator.

**Limitations:**

- The order selection for prediction with kernel and DNN estimators is not trivial.
- To get a complete theory, the “minimal assumption” needs to be verified.
- Although the training of DNN on pseudo data can be parallelized, the computation is still heavy in terms of memory.
- The prediction inference of the general model has not been explored in our approach.

# Thank you!

# References


Bauer, B. and Kohler, M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, 47(4).

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Franke, J., Kreiss, J.-P., Mammen, E., and Neumann, M. H. (2002). Properties of the nonparametric autoregressive bootstrap. *Journal of Time Series Analysis*, 23(5):555–585.

Franke, J., Neumann, M. H., and Stockis, J.-P. (2004). Bootstrapping nonparametric estimators of the volatility function. *Journal of Econometrics*, 118(1-2):189–218.

Guo, M., Bai, Z., and An, H. Z. (1999). Multi-step prediction for nonlinear autoregressive models based on empirical distributions. *Statistica Sinica*, pages 559–570.

Kohler, M., Krzyzak, A., and Langer, S. (2019). Estimation of a function of low local dimensionality by deep neural networks. *arXiv preprint arXiv:1908.11140*.

Pemberton, J. (1987). Exact least squares multi-step prediction from nonlinear autoregressive models. *Journal of Time Series Analysis*, 8(4):443–448.

Politis, D. N. (2015). *Model-free prediction in regression*. Springer.

Politis, D. N. (2024). Scalable subsampling: computation, aggregation and inference. *Biometrika*, 111(1):347–354.

# References


Politis, D. N., Romano, J. P., and Wolf, M. (1999). Subsampling in the iid case. In *Subsampling*, pages 39–64. Springer.

Wolf, M. and Wunderli, D. (2015). Bootstrap joint prediction regions. *Journal of Time Series Analysis*, 36(3):352–376.

Xu, C. and Xie, Y. (2021). Conformal prediction interval for dynamic time-series. In *International Conference on Machine Learning*, pages 11559–11569. PMLR.

Yarotsky, D. (2018). Optimal approximation of continuous functions by very deep relu networks. In *Conference on learning theory*, pages 639–649. PMLR.

# Complete assumptions A1 to A3

- A1. The target mean function $f_0$ lies in the Hölder space $C^{k,\alpha}\left([-B_n, B_n]^d\right)$, which is the space of $k$ times continuously differentiable functions on $[-B_n, B_n]^d$ having a finite norm defined by

$$\|f\|_{C^{k,\alpha}([-B_n,B_n]^d)} = \max\left\{ \max_{\boldsymbol{k}:|\boldsymbol{k}|\leq k} \max_{\boldsymbol{x}\in[-B_n,B_n]^d} \left|D^{\boldsymbol{k}} f(\boldsymbol{x})\right|, \max_{\boldsymbol{k}:|\boldsymbol{k}|=k} \sup_{\substack{\boldsymbol{x},\boldsymbol{y}\in[-B_n,B_n]^d \\ \boldsymbol{x}\neq\boldsymbol{y}}} \frac{\left|D^{\boldsymbol{k}} f(\boldsymbol{x}) - D^{\boldsymbol{k}} f(\boldsymbol{y})\right|}{\|\boldsymbol{x}-\boldsymbol{y}\|^{\alpha}} \right\}.$$

  where the smoothness order is $r = k + \alpha$ with an integer $k \geq 0$ and $0 < \alpha \leq 1$. In particular, we define the function space $\mathcal{F}_{r,d,[-B_n,B_n]} := \{f : \|f\|_{C^{k,\alpha}([-B_n,B_n]^d)} < \infty \text{ and } \|f\|_\infty \leq B_n\}$ and $f_0 \in \mathcal{F}_{r,d,[-B_n,B_n]}$.

- A2. The whole series $\{Y_t\}$ is strictly stationary and geometrically $\beta$-mixing, i.e., $\beta(j) \leq C'_\beta e^{-C_\beta j}$ for some $C_\beta, C'_\beta > 0$. In addition, we denote the marginal stationary distribution of $\{Y_t\}$ by $\mathbb{P}_\pi$ and we assume it is absolutely continuous.

- A3. $\lim_{n\to\infty} \log(n) \cdot \mathbb{P}_\pi\left(\max_{t\in\{1,\ldots,n\}} |Y_t| \geq B_n\right) = 0$.

# Complete Theorem 1

## Theorem 1: $L_2$ error bound of DNN

Under assumptions A1 to A3, taking $B_n \asymp n^{K_B}$ for $K_B \in [0, \frac{1}{2(r+d)+2})$, and setting $L_n \asymp \log(n)$ and $H_n \asymp n^{\left(\frac{d}{r+d}\right)(1/2-K_B)} \log(n)$. Then, if the DNN estimator satisfies $Q_n\left(\hat{f}_n\right) \leq \inf_{f \in \mathcal{F}_n} Q_n(f) + \theta_n$; where $Q_n(\cdot) := \frac{1}{n}\sum_{t=1}^{n}(Y_n - f(\boldsymbol{Y}_{t-1}))^2$, and $\epsilon_n = C\left(n^{rK_B - \frac{r}{r+d}(1/2-K_B)} \log^4(n) + \sqrt{\mu_n + \theta_n}\right)$, where $\mu_n := max\left\{6\mathbb{E}\left[Y_t^2 \mathbb{1}_{\{|Y_t| \geq B_n\}}\right], n^{-1}\right\}$ and $C > 0$ is a constant independent of $n$, then for all $n$ sufficiently large s.t., $n \geq \max\{5, 2Pdim\left(\mathcal{F}_n\right), 16B_n^2/\log(n)\}$ and

$$\max\left\{2, \frac{\tilde{\epsilon}_n(\log n)(\log\log n)}{B_n}\right\} < a := \left\lceil \log^2(n) \right\rceil \leq n/2,$$

where $\tilde{\epsilon}_n = O((2B_n)^r\, n^{-\left(\frac{r}{r+d}\right)(1/2-K_B)})$ shown in the proof. Then, we have

$$P\left(\left\|\hat{f}_n - f_0\right\|_{\mathcal{L}^2} \leq \epsilon_n\right),$$

with probability at least

$$1 - e^{-C_\delta n^{2rK_B+2\left(\frac{d}{r+d}\right)(1/2-K_B)} \log^6(n)} - 2\log(n)\left[\frac{C_\beta' n^{1-C_\beta \log n}}{\log^2(n)} + 2P\left(\max_{t\in\{1,\ldots,n\}} Y_t \geq 2B_n\right)\right];$$

$C_\delta$ is another constant independent with $n$.

# Pertinent prediction interval

The success of the forward bootstrap is due to the fact that we can get a *pertinent* PI (PPI) which is able to capture the model estimation variability.

**Key components of PPI**:

- $\sup_a |\widehat{F}_\epsilon(a) - F_\epsilon(a)| \xrightarrow{p} 0$.
- $\sup_a |\mathbb{P}(\tau_T A_m^* \leq a) - \mathbb{P}(\tau_T A_m \leq a)| \xrightarrow{p} 0$,

For example, we assume that $G(\boldsymbol{X}_{t-d}, \epsilon_t)$ can be decomposed as $m(\boldsymbol{X}_{t-d}) + \epsilon_t$; $A_m^* = \widehat{m}^*(\boldsymbol{x}) - \widehat{m}(\boldsymbol{x})$; $A_m = \widehat{m}(\boldsymbol{x}) - m(\boldsymbol{x})$.

**Remark**: Similar preliminary conditions are assumed in conformal prediction on time series; see Xu and Xie (2021) for example.

# Remark of the main algorithm

**Remark:**

- This algorithm provides a framework to do predictions when model information is unknown. It can directly work for the cases where $G(\cdot, \cdot)$ is in a parametric or non-parametric form.
- Re-defining $\{X^*_{n-d+1} = X_{n-d+1}, \ldots, X^*_n = X_n\}$ is necessary since it exactly mimics the prediction process in the real-world.
- The model estimation variability is captured due to the approximation of $X_{n+j} - \widehat{X}_{n+j}$ by $X^*_{n+j} - \widehat{X}^*_{n+j}$.