# Autoregressive Models I

- LLM is an autoregressive model, so before giving details of LLM, we discuss basic concepts of autoregressive models.
- Autoregressive models predict the next component in a sequence by using information from previous inputs in the same sequence.
- A typical example is time series prediction with applications in stock index prediction, electricity load prediction, etc.

# Autoregressive Models II

- Assume our sequence is

$$z_1, z_2, \ldots .$$

- The way to train a model is by using data shown in the following table.

| training instance | target value |
| --- | --- |
| $z_1, \ldots, z_T$ | $z_{T+1}$ |
| $z_1, \ldots, z_{T+1}$ | $z_{T+2}$ |
| $\vdots$ | $\vdots$ |

# Autoregressive Models III

- In practice, data points occurred long time ago may not be important. We can discard them to make training instances have the same number of values:

| training instance | target value |
| --- | --- |
| $z_1, \ldots, z_T$ | $z_{T+1}$ |
| $z_2, \ldots, z_{T+1}$ | $z_{T+2}$ |
| $\vdots$ | $\vdots$ |

# LLM Is an Autoregressive Model I

- The next-token prediction of LLM is a case of auto-regressive settings.
- Recall we have the setting shown in the following figure.

# LLM Is an Autoregressive Model II

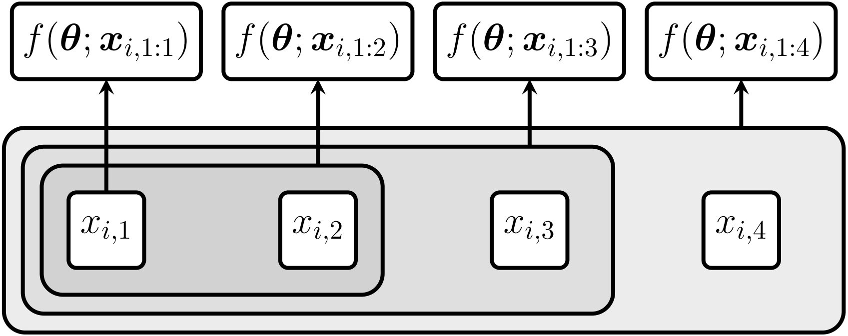


Figure: A sequence of next-token predictions

# LLM Is an Autoregressive Model III

- Note that we aim to have

$$\begin{aligned} f(\boldsymbol{\theta}; \boldsymbol{x}_{i,1:1}) &\approx x_{i,2} \\ f(\boldsymbol{\theta}; \boldsymbol{x}_{i,1:2}) &\approx x_{i,3} \\ &\vdots \end{aligned}$$

- For LLM, the $f$ function is complicated.
- Thus, we begin with learning how to train a simple auto-regressive model.
- From the discussion, we will identify important properties to be used for LLM training/prediction.

# Training a Simple Autoregressive Model I

- Assume we have the following sequence of data

$$z_1, z_2, \ldots$$

  and would like to construct a model for one-step ahead prediction.
- From the observed data, we collect the following (instance, target value) pairs

$$\begin{array}{ll} \boldsymbol{x}_1 = [z_1, \ldots, z_T]^\top & y_1 = z_{T+1} \\ \boldsymbol{x}_2 = [z_2, \ldots, z_{T+1}]^\top & y_2 = z_{T+2} \\ \vdots & \vdots \end{array}$$

# Training a Simple Autoregressive Model II

- Assume we have collected $n$ training instances.
- We can then solve a simple least-square regression problem to get a model

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} (y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2. \tag{1}$$

- Here $\boldsymbol{w}$ includes the model weights.
- We notice two important properties here.
- The first property is that we use matrix operations to handle all data together.

# Training a Simple Autoregressive Model III

- Specifically, (1) has an analytic solution:

$$\text{optimal } \boldsymbol{w} = (X^\top X)^{-1} X^\top \boldsymbol{y},$$

  where

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \text{ and } X = \begin{bmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_n^\top \end{bmatrix} \in \mathbf{R}^{n \times T}.$$

- For simplicity, we assume that $X^\top X$ is invertible.

# Training a Simple Autoregressive Model IV

- We see that even though $y_{T+1}$ is the target value of the first instance, it is also a feature of the second training instance.
- Our setting allows the model building by efficient matrix operations.
- That is, we handle all training data together, even though there are some auto-regressive relationships between them.
- The reason we can do this is because our prediction function on training data is the same as the one we use for future prediction.

# Training a Simple Autoregressive Model V

- In testing, for a vector $\boldsymbol{x}$ containing past information, we use $\boldsymbol{w}^\top \boldsymbol{x}$ to get our prediction.
- In training, for any $\boldsymbol{x}_i$, in (1) we use the same way to hope that $\boldsymbol{w}^\top \boldsymbol{x}_i$ is close to $y_i$.
- This is the second crucial property we will use in our LLM design.