1. **ARIMA**
   1. **What is ARIMA ?** 
      1. **Definition**

The Autoregressive Integrated Moving Average Model, or ARIMA for short is a standard statistical model with:

* **AR**: *Autoregression*. A model that uses the dependent relationship between an observation and some number of lagged observations.
* **I**: *Integrated*. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
* **MA**: *Moving Average*. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

It is used in statistics and econometrics to measure events that happen over a period of time. The model is used to understand past data or predict future data in a series. It’s used when a metric is recorded in regular intervals, from fractions of a second to daily, weekly or monthly periods.

* + 1. **ARIMA Parameters**

Each component in ARIMA functions as a parameter with a standard notation. For ARIMA models, a standard notation would be ARIMA with p, d, and q, where integer values substitute for the parameters to indicate the type of ARIMA model used. The parameters can be defined as:

* *p*: the number of lag observations in the model, also known as the lag order.
* *d*: the number of times the raw observations are differenced; also known as the degree of differencing.
* q: the size of the moving average window, also known as the order of the moving average.
  1. **How to calculate ARIMA**

ARIMA model can be used in a lot of tools like R, Excel, Python,… So to start the model:

First step is to gather the data need.

After that we need to a unit root test with ADF Extended Unit Testing Model to determine whether or not it is stationary .

Diagram

Description automatically generated with medium confidence

With :

yt : The series of metrics over time under consideration

k : Delay length

: White noise

This will conduct testing in both absent and time-oriented cases using model after model

The results of the ADF test are often very sensitive to the choice of delay length k, so the AIC (Akaike's Information Criterion) standard of Akaike (1973) was used to select the optimal k for the ADF model. In particular, the k-value is selected so that the AIC is the smallest.

In case p-value < 0.05, we will discard the null hypothesis, accept the substitution hypothesis. Then we can assert that the series have no unit test and have stationary .

Text

Description automatically generated with low confidence

Next step is to calculate AR(p) and MA (q)

At this step we will use the ACF of (**Autocorrelation Function) to calculate q parameter and PACF (Partial Autocorrelation Function) to calculate p parameter to choose which model will be use**

**ACF :**

A picture containing text

Description automatically generated

**With :** Text

Description automatically generated

yt : The time series stops at t

yt+k : The time series stops at t + k

^ : mean of stationary

rk : Correlation value between yt and yt+k  at k latency

if rk = 0 there won’t be autocorrelation

**PACF**

Text

Description automatically generated with medium confidence

With :

rk : autocorrelation at k latency

v : variance

Ckj : autocorrelation at k latency, determinate the effects of latencies

**AR(p) model**



With:

y(t) : current value of the time series

y(t-1), y(t-2), ... : past value of the time series

a0, a1, a2, …: Regression analysis parameters

et  : random forecast error of the current period. The average value is expect to be 0

The stop condition is the sum of the regression analysis parameters is smaller than 1



**MA (q) model**



With:

y(t) : past value of the time series

e(t) : random forecast error, ti value is unknown and it mean is 0

e(t-1), e(t-2), ... : random past forecast error

b0, b1, b2, ...: mean of y(t) and the mobile average coefficients.

q : error being used in average coefficients.

The necessary condition is that the sum of the mobile average coefficients must be less than 1.



ARIMA model (mix model)

Text

Description automatically generated

With :

y(t) : current value of the time series

y(t-p), và e(t-q) : past value and past forecast error of the time series

a0, a1, a2, ..., b1, b2, ... : Regression analysis coefficients

Next, determine the order of regression (p) and order of moving average (q) by comparing autocorrelations and partial autocorrelations.

Finally choose the model

Graphical user interface, text, application, email

Description automatically generated

* 1. **Why choosing ARIMA ?**
* ARIMA models have strong points and are good at forecasting based on past circumstances.
* Good for short-term forecasting
* Only needs historical data
* Models non-stationary data

1. **ARIMAX**
   1. **What is ARIMAX ?**
      1. **What is ARIMA ?**

The Autoregressive Integrated Moving Average Model, or ARIMA for short is a standard statistical model with:

* **AR**: *Auto-regressive*: A model that uses the dependent relationship between an observation and some number of lagged observations.
* **I**: *Integrated*. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
* **MA**: *Moving Average*. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

It is used in statistics and econometrics to measure events that happen over a period of time. The model is used to understand past data or predict future data in a series. It’s used when a metric is recorded in regular intervals, from fractions of a second to daily, weekly or monthly periods.

* + - * 1. **What is X stand for ?**

X or Exogenous variable is a factor in the model whose value is identified by the factors of variables outside the model study .

**What is ARIMAX exactly is ?**

An Autoregressive Integrated Moving Average with Explanatory Variable (ARIMAX) model can be viewed as a multiple regression model with one or more autoregressive (AR) terms and/or one or more moving average (MA) terms. This method is suitable for forecasting when data is stationary/non stationary, and multivariate with any type of data pattern, i.e., level/trend /seasonality/cyclicity.

ARIMAX is related to the ARIMA technique but, while ARIMA is suitable for datasets that are univariate

ARIMAX is suitable for analysis where there are additional explanatory variables (multivariate) in categorical and/or numeric format

1. **RNN**
   1. **What is RNN**

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes can create a cycle, allowing output from some nodes to affect subsequent input to the same nodes [1]. RNN works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer[2].

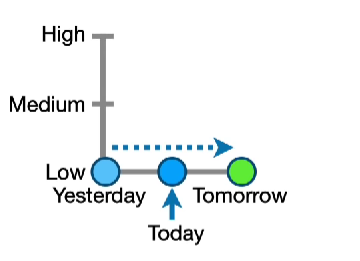
* 1. **How RNN**

Example:

The Low price has the value 0

The Medium has the value 0.5

The High has the value 1



With feedforward neural networks, I could input yesterday value and let the networks predict the value for today.

Ảnh có chứa biểu đồ

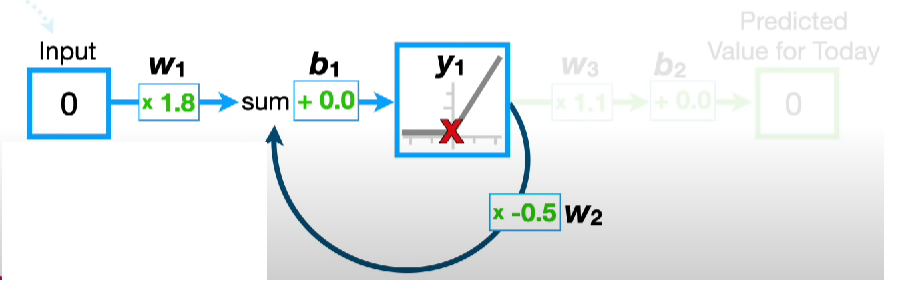
Mô tả được tạo tự động

(X1 \* W1) + b1 = (0 \* 1.8) + 0 = 0

Y1 =f(x) = max(0,0) = 0

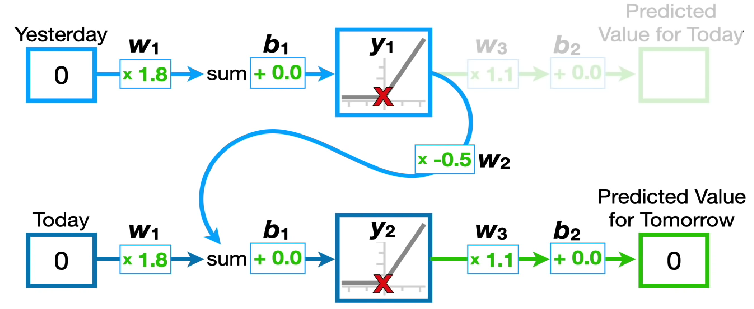
Output = (Y1 \* W3) + b2 = (0 \* 1.1) + 0 = 0

But if I’ve already known the value of today and I want to know the value of tomorrow base on the value of yesterday and today. The RNN comes and solve this problem.



Yesterday’s value

After input the yesterday’s value, the network calculates as normal. After calculates, instead giving us the predict for today’s value, the network gives feedback to input layer and the network continue to input today’s value.



(X1 \* W1) + b1 = (0 \* 1.8) + 0 = 0

Y1 =f(x) = max(0,0) = 0

(X2 \* W1) + (Y1 \* W2) + b1 = (0 \* 1.8) + (0 \* -0.5) + 0 = 0

Y2 = f(x) = max(0,0) = 0

Output = (Y2 \* W3) + b2 = (0 \* 1.1) + 0 = 0 (Tomorrow’s value)

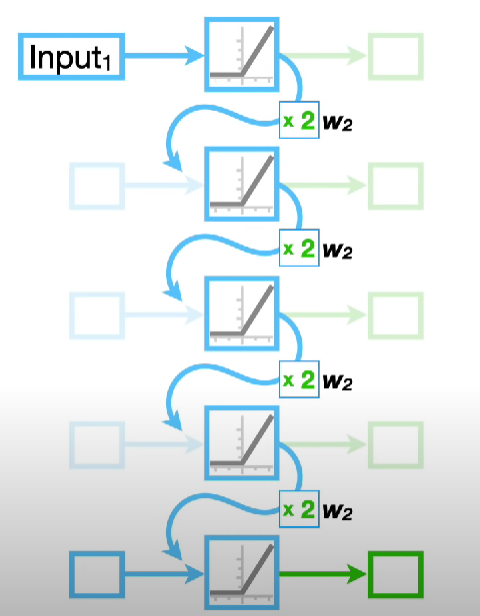
To predict the value of tomorrow base on the value of yesterday and today’s value, I must unroll 2 times. If I want to predict the value of tomorrow base on 50 days, I must unroll 50 times. Each time the network unroll, the weights and biases are shared across every inputs and no matter how many times we unroll a recurrent neural network, we never increase the number of weights and biases that we have to train.

* 1. **Problem**

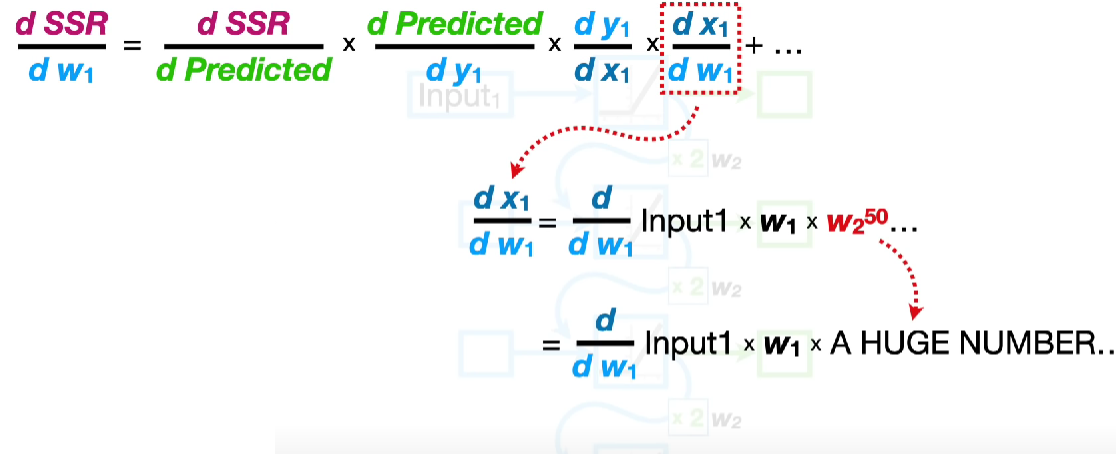
The problem of recurrent neural network is the more we unroll, the harder it is to train.

To train the network, we have to modify weights and biases to keep the loss value at lowest at possible.

Let take a look on weight W2.



If we unrolled the recurrent network 4 times, we must multiply the input value by W2 raise to the number of times we unrolled.

If we unroll 50 times, this can cause a problem called Exploding Gradient

Ảnh có chứa biểu đồ

Mô tả được tạo tự động

One way to prevent the exploding gradient would be limit W2 lower then value 1.

How ever, this results in the vanishing gradient.

1. **LSTM**
   1. **What is LSTM?** [3]

LSTM stands for long short-term memory networks, used in the field of Deep Learning. It is a variety of recurrent neural networks (RNNs) that are capable of learning long-term dependencies, especially in sequence prediction problems. LSTM has feedback connections, i.e., it is capable of processing the entire sequence of data, apart from single data points such as images. This finds application in speech recognition, machine translation, etc. LSTM is a special kind of RNN, which shows outstanding performance on a large variety of problems.

* 1. **How LSTM work?** [4]

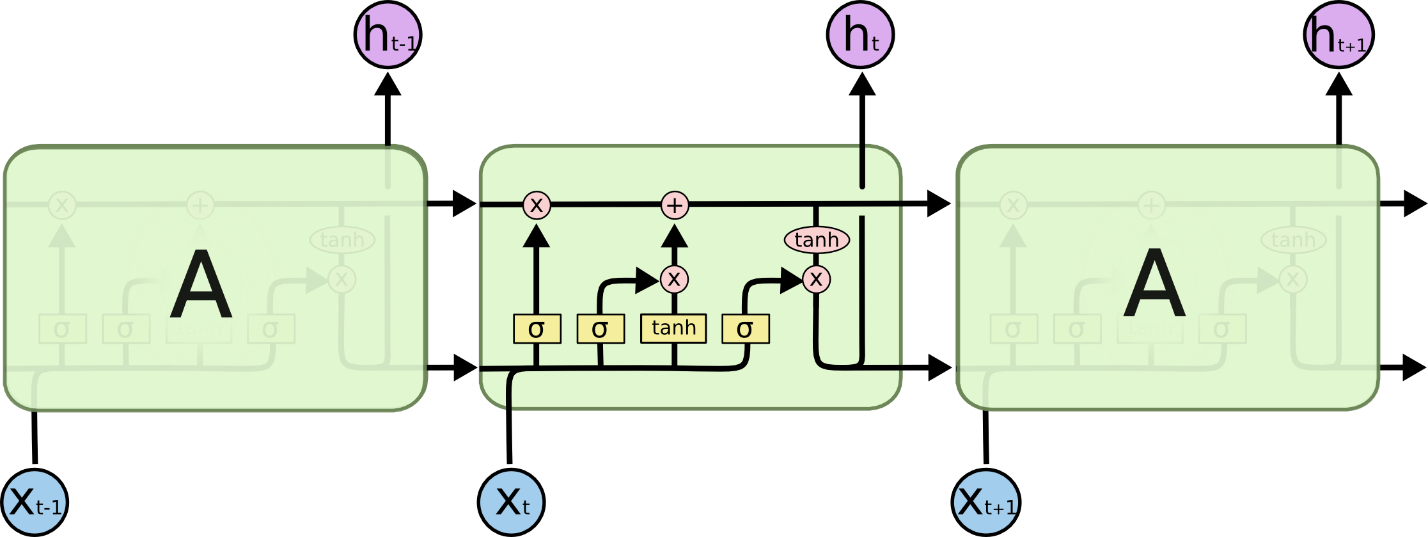
LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

Diagram

Description automatically generated

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.



Don’t worry about the details of what’s going on. We’ll walk through the LSTM diagram step by step later. For now, let’s just try to get comfortable with the notation we’ll be using.

Diagram

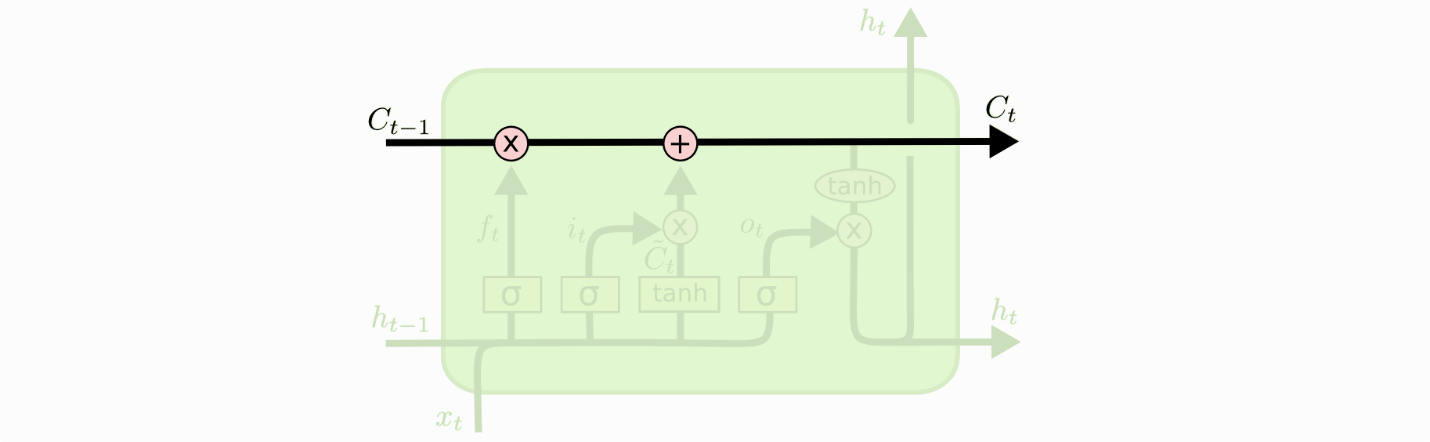
Description automatically generated

In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

**The Core Idea Behind LSTMs**

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.

The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It’s very easy for information to just flow along it unchanged.



The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.

Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.

Graphical user interface, application

Description automatically generated

The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!”

An LSTM has three of these gates, to protect and control the cell state.

**Step-by-Step LSTM Walk Through**

The first step in our LSTM is to decide what information we’re going to throw away from the cell state. This decision is made by a sigmoid layer called the “forget gate layer.” It looks at  and , and outputs a number between 0 and 1 for each number in the cell state . A 1 represents “completely keep this” while a 0 represents “completely get rid of this.”

Let’s go back to our example of a language model trying to predict the next word based on all the previous ones. In such a problem, the cell state might include the gender of the present subject, so that the correct pronouns can be used. When we see a new subject, we want to forget the gender of the old subject.

Diagram

Description automatically generated

The next step is to decide what new information we’re going to store in the cell state. This has two parts. First, a sigmoid layer called the “input gate layer” decides which values we’ll update. Next, a tanh layer creates a vector of new candidate values, , that could be added to the state. In the next step, we’ll combine these two to create an update to the state.

In the example of our language model, we’d want to add the gender of the new subject to the cell state, to replace the old one we’re forgetting.

Diagram

Description automatically generated

It’s now time to update the old cell state, , into the new cell state . The previous steps already decided what to do, we just need to actually do it.

We multiply the old state by , forgetting the things we decided to forget earlier. Then we add \* . This is the new candidate values, scaled by how much we decided to update each state value.

In the case of the language model, this is where we’d actually drop the information about the old subject’s gender and add the new information, as we decided in the previous steps.

Diagram

Description automatically generated with medium confidence

Finally, we need to decide what we’re going to output. This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we’re going to output. Then, we put the cell state through tanhtanh (to push the values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

For the language model example, since it just saw a subject, it might want to output information relevant to a verb, in case that’s what is coming next. For example, it might output whether the subject is singular or plural, so that we know what form a verb should be conjugated into if that’s what follows next.

A picture containing text, clock

Description automatically generated

**Why do we use LSTM?**

One of the key advantages of using LSTM networks lies in the fact that they address the vanishing gradient problem that makes network training difficult for a long sequence of words or integers. Gradients are used for updating RNN parameters and for a long sequence of words or integers; these gradients become smaller and smaller to the extent that, effectively, no network training can take place. LSTM networks help to overcome this problem and make it possible to capture long-term dependencies between keywords or integers in sequences that are separated by a large distance. [5]

We use LSTM because it helps us to better understand and predict the behavior of stock markets. It can also be used to identify patterns in data that may not be immediately obvious. By understanding these patterns, we can make more informed decisions about when to buy or sell stocks. In addition, LSTM can help us to filter out noise from data, making it easier to identify Trends. [6]

**Reference:**

[1] “Recurrent neural network,” *Wikipedia*. Apr. 07, 2023. Accessed: May 02, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Recurrent\_neural\_network&oldid=1148729712

[2] “Recurrent Neural Network (RNN) Tutorial: Types and Examples [Updated] | Simplilearn,” *Simplilearn.com*. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn (accessed May 02, 2023).

[3] K. L. says, “What is LSTM - Introduction to Long Short Term Memory,” *Intellipaat Blog*, May 28, 2020. https://intellipaat.com/blog/what-is-lstm/ (accessed Apr. 26, 2023).

[4] “Understanding LSTM Networks -- colah’s blog.” https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed Apr. 26, 2023).

[5] “Why do we use LSTM networks? | Advanced Deep Learning with R,” *Packt*. https://subscription.packtpub.com/book/data/9781789538779/15/ch15lvl1sec76/why-do-we-use-lstm-networks (accessed Apr. 26, 2023).

[6] “A Guide to Long Short Term Memory (LSTM) Networks.” https://www.knowledgehut.com/blog/web-development/long-short-term-memory (accessed Apr. 26, 2023).