Convolutional Neural Network (CNN)

It is one of the basic tools for deep learning, which falls under the umbrella of deep neural networks (DNN), which includes another type of DNN, which is recursive neural networks (RNN). The CNN method is sometimes called a deep convolutional neural network (Deep-CNN) when it is a multi-layer network containing more than two layers. Since the CNN method currently contains two or more layers in its structure, the terms CNN and Deep-CNN have become the same scientific concept (Theobald, 2017). The CNN structure consists of two main parts: the feature recognition layer, in which convolution and pooling operations are performed to recognize image features such as edges and color gradation, and the fully connected layer, which receives the outputs of the feature recognition layer as inputs to perform the classification process, as shown in Figure 5 below.

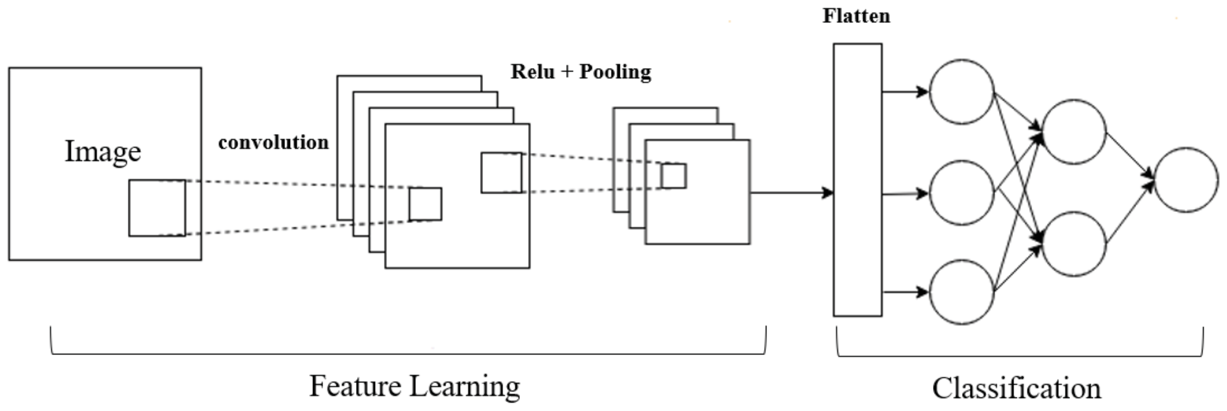


Figure 5: Convolutional neural network architecture

First, the appropriate CNN structure is chosen by determining the number of layers. The layers consist of the input layer, the hidden layers, and the output layer. As for the input layer, it is determined by the number of inputs, which correspond to the concept of explanatory variables in the Multiple Linear Regression model, or called the autoregressive variables of the time series. As for the size of the hidden layer, it is determined by the number of neurons contained in the layer, which is often more than the number of inputs. As for the output layer, the number of outputs corresponds to the number of classification options required (Gamboa, 2017). In the first hidden layer, the number of filters is determined, symbolized by (m), and the size of each filter is symbolized by (f), and each filter contains the initial values of the weights, symbolized by (w), and these filters wrap around all the points of the image to recognize the edges, vertical or horizontal lines, or other angles of the image and the details that make up the image. As for the stride step that the filter moves (meaning the number of pixels that are skipped), if not specified, it will default to one step for the filter towards the left of the image. When the filter wraps around the image, the size of the image will shrink and part of the data that makes up the image will be lost. To solve this problem, padding is added with rows and columns along the length, width and perimeter of the image. The size of the matrix resulting from the process of wrapping each filter around the image is done through a simplified mathematical process as shown in equation (4) below.

$$I \times J = \left[\left(\frac{H + 2p - f}{s}\right) + 1\right] \times \left[\left(\frac{W + 2p - f}{s}\right) + 1\right] \tag{4}$$

Where (H): image length.

(W): image width.

(P): padding.

(f): one of the filter dimensions.

(s): step.

Then the bias value of each filter is added to the corresponding matrix elements as in equation (5) below, whose outputs can be called the outputs of the additive function.

$$SUM_{IJk} = \sum_{i=1}^{f_2}\sum_{j=1}^{f_1} w_{f_1 f_2} x_{HW} + b_k \qquad (5)$$

Where i, j represent the dimensions or number of rows and columns in each image respectively. And f1, f2 represent the number of rows and columns in each filter respectively. And k = 1, 2,…., m which symbolizes the sequence of the output of each filter. One of the transformation functions is applied to Equation (5). The most commonly used transformation functions in neural networks are as follows.

1. Logistic Sigmoid function:

$$f\left(\text{SUM}\right) = \frac{1}{1+e^{-(\text{SUM})}} \qquad (6)$$

SUM represents the inputs of the transformation function, which represents the outputs of the additive function and generates its outputs within the limits (0,1) as shown in Figure 6 below.

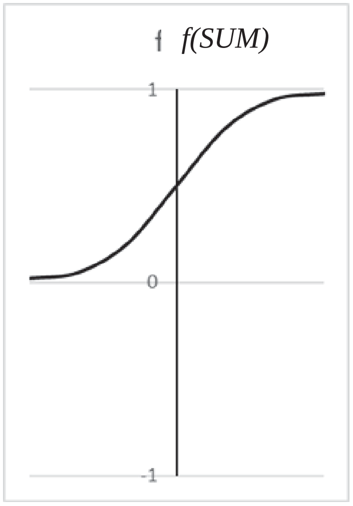


Figure 6: Logistic function

2. Tan sigmoid function

$$f(\text{SUM}) = \frac{2}{1+e^{-2(\text{SUM})}} - 1 \tag{7}$$

SUM represents the inputs of the transformation function, which represents the outputs of the additive function and generates its outputs within the limits (-1,1) as shown in Figure 7 below.

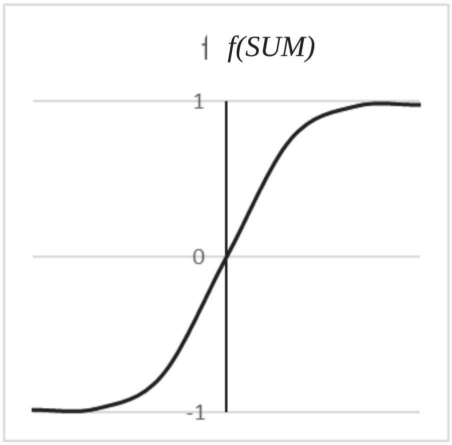


Figure 7: Tan function

3. Pure Line function

$$f(\text{SUM}) = \text{SUM} \tag{8}$$

SUM represents the inputs of the transformation function, which represents the outputs of the additive function and generates its outputs within the limits (-1,1) as shown in Figure 8 below.

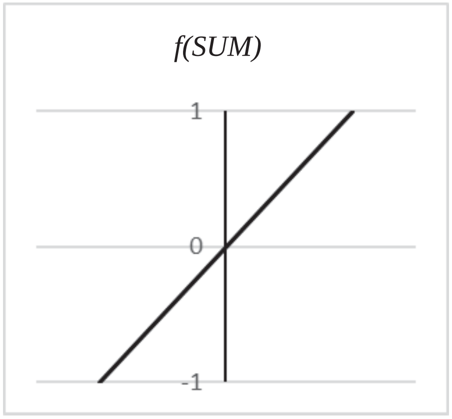


Figure 8: Linear function

4. Rectified Linear Unit function (ReLU)

$$f(\text{SUM}) = \begin{cases} 0 & \text{SUM} \leq 0 \\ x & \text{SUM} > 0 \end{cases} \tag{9}$$

SUM represents the inputs of the transformation function, which represents the outputs of the additive function and generates its outputs greater than or equal to (0), as shown in Figure 9 below.

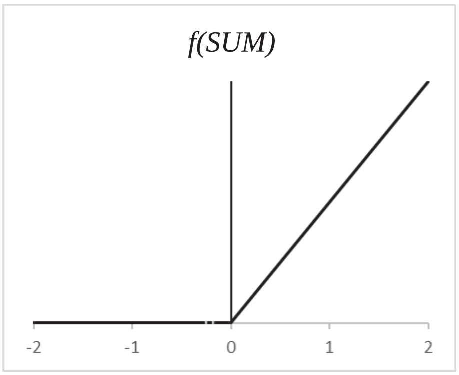


Figure 9: ReLU function

5. Softmax function

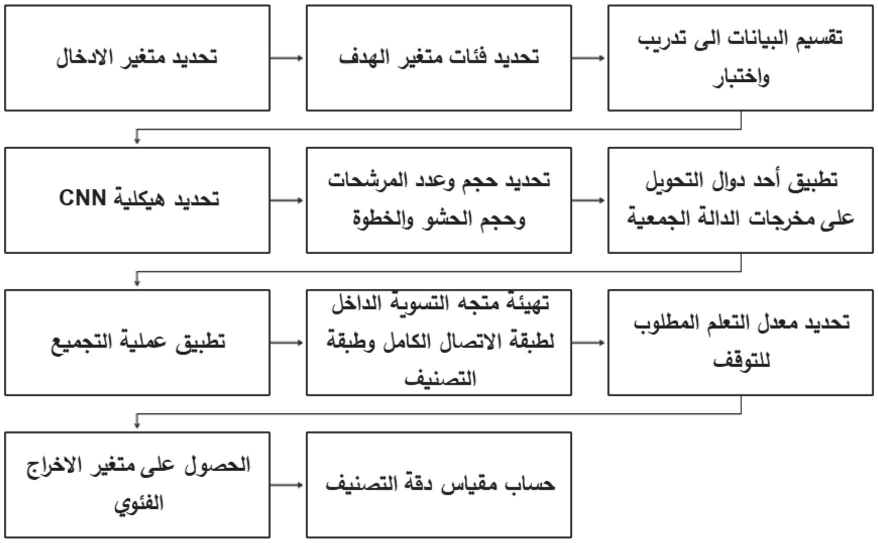
$$f\left(\mathrm{x_i}\right) = \frac{exp(\mathrm{x_i})}{\sum_{\mathrm{j=1}}^{\mathrm{k}} exp(\mathrm{x_j})} \tag{10}$$

The outputs $\mathrm{x_i}$ of the fully connected layer, i.e. in binary classification, represent the probability of success or failure $\mathrm{x_j}$ for each output, representing the sum of the probabilities. This function is applied to the outputs of the fully connected layer and the result is probabilities according to the number of categories of the response variable. These probabilities are positive numbers and each number represents the probability of belonging to a category of the categories and their sum is equal to one. After the process of entering the transformation function on the outputs of the additive function, the pooling process is applied by dividing the matrix into square or rectangular pooling regions, as the matrix is divided into a number of small matrices. This is done by determining their size and number of steps. Then the highest value is taken in the case of applying max pooling or the average of the values in the case of applying average pooling for each region. Thus, each small matrix is reduced to only one value, which resulted in reducing the size of the matrix so that the matrix at the end of the first hidden layer becomes smaller in dimensions. Then it is converted into a vector, which is a single column containing all the properties through a process called flattening. The vector represents the inputs of the fully connected layer and this layer collects all the features learned by the previous layers across the images (Neapolitan & Jiang, 2018). The outputs of the fully connected layer are determined by the number of classification classes depending on the number of classes of the dependent variable and the weights are also determined randomly in each neuron and this layer is often followed in the case of classification by the Softmax transformation function. In the Classification Layer, the error value is calculated as the goal of including the classification layer is to reduce the error by updating the weight values (w) and bias (b) and it is an iterative process until reaching the optimal values as each iteration process obtains certain values of the weights and bias and tests the error rate and this process is called stepwise regression and this procedure is technically called Gradient Descent (GD) which allows determining the direction towards reducing errors and the GD equation is as follows.

$$p_{i+1} = p_i - \alpha \frac{1}{n} \sum_{j=1}^{n} (y_{ij} - \hat{y}_{ij}).x_{ij} \tag{11}$$

Where (α) represents the learning rate, which is a very small value between (0 and 1), and p_i represents w_i or b_i in the current iteration, and p_{i+1}represents w_{i+1} or b_{i+1} in the new subsequent iteration, y_{ij} represents the value of observation j of the target variable in the current iteration i, $\hat{y}_{ij}$ represents the value of observation j of the output variable in the current iteration i of the classification layer, x_{ij} represents observation j of the input variable in the current iteration i (Zhao et al., 2017).

After obtaining the optimal values for weights and bias through the classification layer, the final action performed by the CNN is to classify the time series by comparing the variable $\hat{y}_{\text{final}}$ with the original values variable, which is the target variable. The accuracy of the classification model will be calculated with respect to the actual values of the time series using the classification accuracy metric. Figure 10 below shows the general framework of the CNN algorithm.



تحديد متغير الادخال
تحديد فئات متغير الهدف
تقسيم البيانات الى تدريب واختبار
تحديد هيكلية CNN
تحديد حجم وعدد المرشحات وحجم الحشو والخطوة
تطبيق أحد دوال التحويل على مخرجات الدالة الجمعية
تطبيق عملية التجميع
تهيئة متجه التسوية الداخل لطبقة الاتصال الكامل وطبقة التصنيف
تحديد معدل التعلم المطلوب للتوقف
الحصول على متغير الاخراج الفئوي
حساب مقياس دقة التصنيف

Figure 10: General structure of the convolutional neural network algorithm