# 2.5 Introduction of Deep Learning

Deep learning is part of broader family of machine learning methods based on artificial neural networks with representation learning [1]. Deep learning architectures can be divided into Feed-Forward-Net and Recurrent-Neural-Network. Feed-Forward-Net is an artificial neural network, which consists of input layer, hidden layers and output layer. The information moves in only on direction, from the input node, through the hidden nodes to the output nodes[2]. The basic difference of the Recurrent-Neural-Network from Feed-Forward-Net is that RNN can use intern state (memory) to process the variable length of inputs. The representative architecture is long short-term memory.

In our project, the features extracted from various paper, such as mass, dark gray proportion, are the values with fixed length, not a sequence of inputs. The outputs are the number of the paper class. According to these characters, we selected the architectures, such as Convolutional Neural Network and some pre trained Network, which belong to the Feed-Forward-Network.

## 2.5.1 Convolutional Neural Network

Convolutional neural network is one of the most popular algorithms for deep learning. Most commonly applied to analyzing visual imagery [3]. Figure 1 shows the basic architecture of CNN. Like the most other Neural Network, a CNN is composed of an input layer and an output layer and many hidden layer in between. The typical character of CNN is the mathematical operation called convolution. Convolution is specialized kind of linear operation and it can create feature from input images. In CNN this operation is executed by multiplication of matrix between input image and a series of convolutional filters. The rectified linear unit (ReLU) can transfer the output of each convolutional layer and allow for nonlinearity. The final part in one feature learning is pooling. It can reduce the number of parameters, which should be learned by Network and combine the feature with other pixels together. Normally, these three operation are repeated over tens or hundreds layers in order to detect different features.

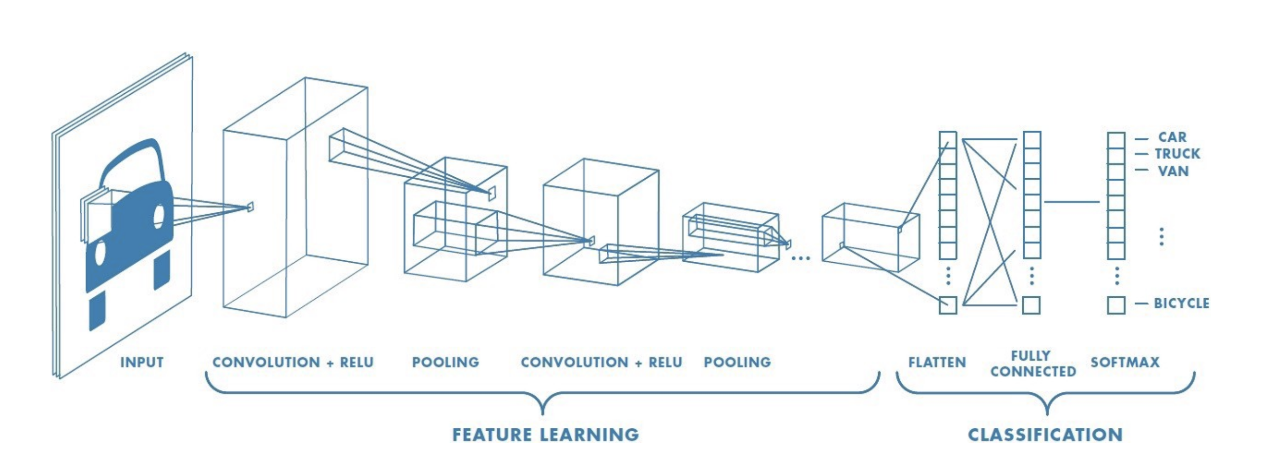


Fig.1 the basic architecture of CNN

After feature detection, the architecture shifts to fully connected layer, which is similar to the traditional multi-layer perceptron neural network (MLP). If the problem is classification, there is also a softmax layer to calculate the probability of the input in each class

In principle CNN is applied in image classification, which belongs to 2-D problem. In our project, the CNN is also planned to apply in NIR spectrum, in order to predict the content of Kaolin. However, the NIR spectrum is a 1-D spectrum. It means that, directly using the CNN is not possible. Chenhao Cui [4]from University College London has proposed a new method about implementation of traditional CNN in spectroscopic analysis by changing the input size of Input layer and build a new series of convolution filters which are suitable for 1-D input data. Malek, Melgani and Bazi [5] also explored 1D-CNN for spectroscopic regressions. In figure 2 shows the converted architecture of a CNN, which can be applied in NIR spectrum. The concrete process included the feature extraction and preprocessing will be introduced in chapter 3.

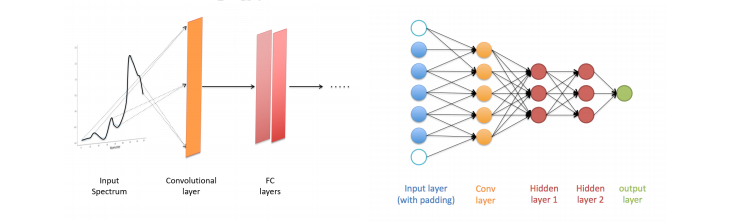


Fig.2 CNN architecture applied NIR spectroscopic analysis [4]

The most significant advantage of CNN compared to traditional machine learning is that it is not necessary for CNN to extract the feature manually. The convolutional filter can detect the feature from computer vision. It can create more reliable features than them from the algorithm written by human.

## 2.5.2 Transfer learning

Although data mining and machine learning technologies have already success in many knowledge engineering areas including classification, regression and clustering[6], for deep learning network, it still needs a large amount of data for training and testing, if a neural network wants to be built for a specify application. Nowadays, in some applications (such as in paper classification), data collection is still a troublesome and expensive work. In such case, Transfer learning is a desirable choice. Instead of building a new neural network, transfer learning can achieve a relative precise result by fine-tuning based on a pre trained network. Nowadays there are various pre trained architectures of neural network. Most of them can be taken in image classification problem.

Since there are totally 3800 objects can be used for paper classification, it is impossible to build a new neural network as classifier. Therefore, it is more property to build a classification neural network by transfer learning. In this chapter, four kinds of typical architecture, which are used in this project will be introduced. They are Alexnet, GoogLenet, VGG-net and ResNet. All the following introduction of neural network are referred to the architecture in Matlab® 2020a

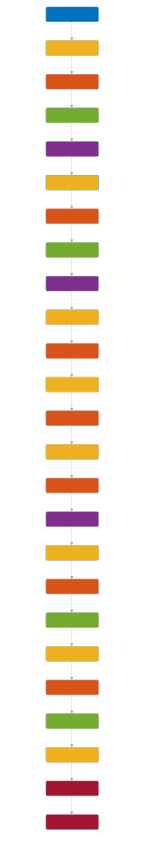
**AlexNet** is a convolutional neural network that is 8 layers deep. Figure 3 shows the overview architecture of AlexNet. It contains 8 leraned layers – five convolutional and three fully-connected.

Fig. 3 Architecture of AlexNet

The output of the last fully connected layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels [7]. The Outstanding of AlexNet compared to the other Neural Network is that it enlarge the learning capacity of CNNs by increasing the depth and breath of a CNNs. Addtionaly, AlexNet uses ReLU Nonlinearity and Local Response Normalization as depicted in Figure 4, in order to reduce the overfitting problem during the training. Therefore, Alexnet has better robustness and compared to the other popular pre trained network such as GoogLeNet, those are also used in this project, it has also a simpler architecture.

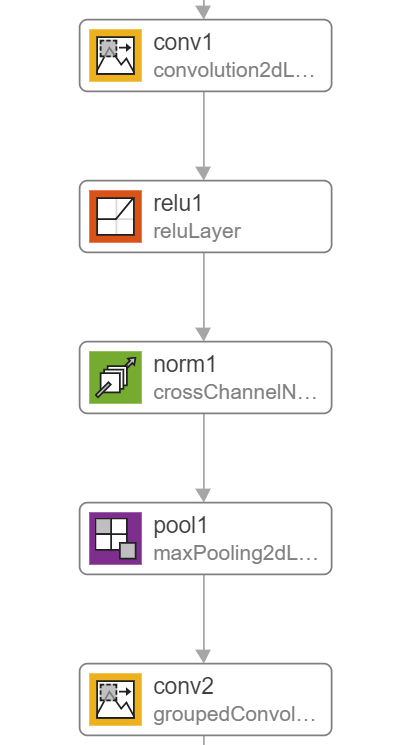


Fig.4 the first feature detection layer in AlexNet

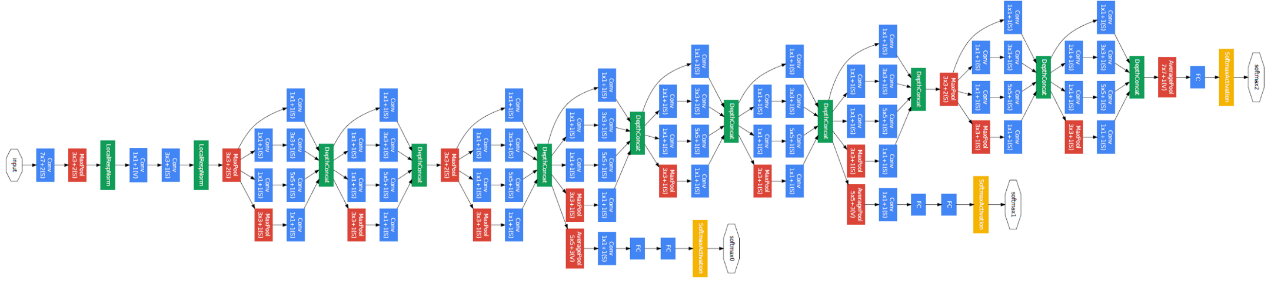
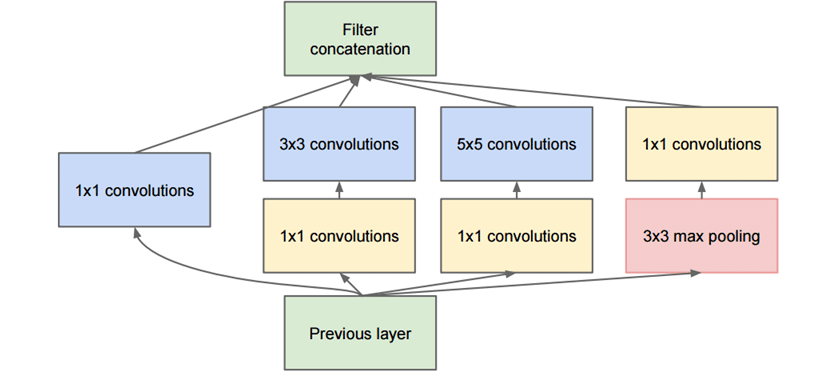
**GoogLeNet** is another convolutional neural network with 22 layers deep. Its architecture is called inception. Figure 5 shows the overview of the GoogLeNet.

Fig 5 Architecture of GoogLeNet

The main idea of the Inception module (Fig. 6) is that it stacks 1x1 convolution for compute reductions before the expensive 3 x 3 and 5 x 5 convolutions [8]. The Function is similar to the PCA, which maintains the height and width of feature map but reduces the depth. The advantage of it is learning efficiency. Besides, compared to Alexnet, GoogLeNet reduced the number of parameters from 60 million to 4 million.

Fig. 6 Inception module

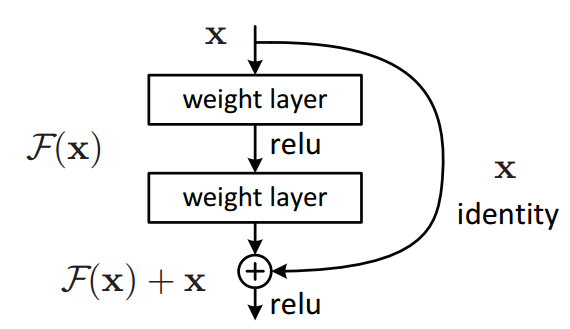
**ResNet** is an artificial neural network and uses residual learning function instead of unreferenced learning function. The hallmark of ResNet is that it can solve the vanish gradient problem in deep layer. The principle of the Residual learning is described in Figure 7. Instead of underlying mapping H(x), Residual mapping F(x) is as Input to fit each layer. In this time, the original feature map H(x) is F(x) + x. This formulation can be realized by feedforward neural network with “shortcut connection” [9].

Fig. 7 Residual learning: a building block

As mentioned before, in AlexNet there are many 11 x 11 and 5 x 5 convolutions. They cause much kernel parameters and decrease the calculation performance. However, a large convolution kernel can enlarge the receptive field and combine more information from image, which is benefited to increase quality of the generated feature maps. GoogLeNet solves this problem by stacking a 1 x 1 convolution to reduce the depth of feature map.

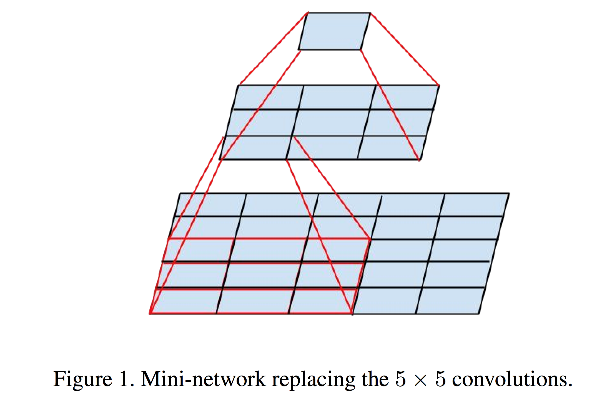
**VGGNet** is also a convolutional neural network, which can also solve this dilemma from another perspective. VGGNet use very small 3 x 3 receptive fields through the whole net [10]. As shown in Figure 8, a 5 x 5 convolution layer can replaced by a stack of two 3 x 3 convolution layers. It confirms that a stack of two 3 x 3 convolution has more effective receptive field than a 5 x 5 convolutions. By analogy three such layers can achieve similar function of a 7 x 7 convolution layer.

Fig. 10 Mini-network replacing the 5 x 5 convolutions

Figure 11 shows the performance of different tranning model according to the accuracy and utilization of computer resource (specially in GPU). It can be clearly concluded that, the