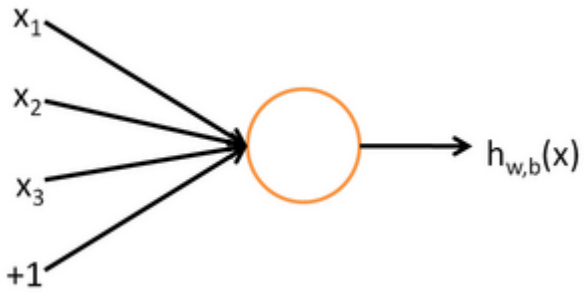
Basic concepts of convolutional neural networks

Inspired by Hubel and Wiesel's research on the cat's visual cortex electrophysiology, someone proposed a convolutional neural network (CNN). Yann Lecun first used CNN for handwritten digit recognition and has maintained its dominance in this problem. In recent years, convolutional neural networks have continued to exert force in multiple directions, and have made breakthroughs in speech recognition, face recognition, general object recognition, motion analysis, natural language processing, and even brain wave analysis.

The difference between convolutional neural networks and ordinary neural networks is that convolutional neural networks include a feature extractor composed of a convolutional layer and a sub-sampling layer. In the convolutional layer of a convolutional neural network, a neuron is only connected to some neighboring neurons. In a convolutional layer of CNN, it usually contains several feature planes (featureMap). Each feature plane is composed of some rectangularly arranged neurons. Accumulate. The convolution kernel is generally initialized in the form of a random decimal matrix. During the network training process, the convolution kernel will learn to obtain reasonable weights. The direct benefit of sharing weights (convolution kernels) is to reduce the connections between layers of the network, while reducing the risk of overfitting. Subsampling is also called pooling, and there are usually two forms of mean subsampling (mean pooling) and maximum subsampling (max pooling). Subsampling can be viewed as a special convolution process. Convolution and subsampling greatly simplifies model complexity and reduces model parameters.

2.Principles of convolutional neural networks

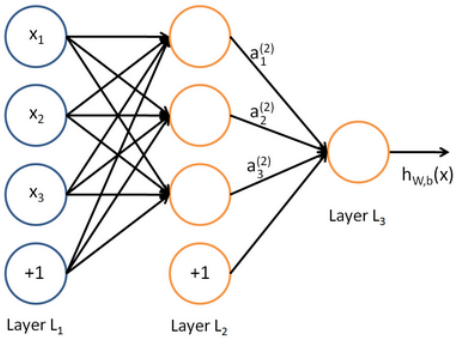
First introduce the neural network. For details of this step, refer to resource 1. Brief introduction. Each unit of the neural network is as follows:



The corresponding formula is as follows:

$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$$

Among them, this unit can also be called Logistic regression model. When multiple units are combined and have a hierarchical structure, a neural network model is formed. The figure below shows a neural network with a hidden layer.



The corresponding formula is as follows:

$$
\begin{aligned}
a_1^{(2)} &= f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \\
a_2^{(2)} &= f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}) \\
a_3^{(2)} &= f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}) \\
h_{W,b}(x) &= a_1^{(3)} = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})
\end{aligned}
$$

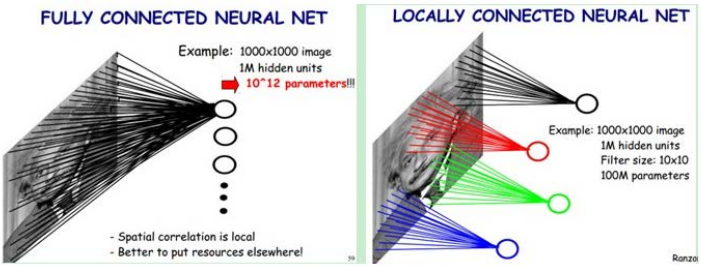
More similar, it can be extended to have 2, 3, 4, 5, ... hidden layers.

The difference between convolutional neural networks and ordinary neural networks is that convolutional neural networks include a feature extractor composed of a convolutional layer and a sub-sampling layer. In the convolutional layer of a convolutional neural network, a neuron is only connected to some neighboring neurons. In a convolutional layer of CNN, it usually contains several feature planes (featureMap). Each feature plane is composed of some rectangularly arranged neurons. Accumulate. The convolution kernel is generally initialized in the form of a random decimal matrix. During the network training process, the convolution kernel will learn to obtain reasonable weights. The direct benefit of sharing weights (convolution kernels) is to reduce the connections between layers of the network, while reducing the risk of overfitting. Subsampling is also called pooling, and there are usually two forms of mean subsampling (mean pooling) and maximum subsampling (max pooling). Subsampling can be viewed as a special convolution process. Convolution and subsampling greatly simplifies model complexity and reduces model parameters.

A convolutional neural network consists of three parts. The first part is the input layer. The second part consists of a combination of n convolutional layers and pooling layers. The third part consists of a fully connected multilayer perceptron classifier.

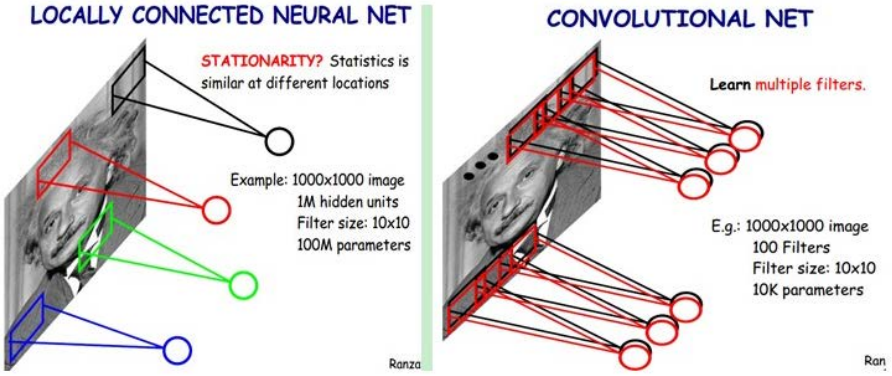
There are two ways to reduce the number of parameters in a convolutional neural network. The first method is called a local perceptual field. It is generally believed that people's perception of the outside world is from local to global, and the spatial relationship of the image is also close to the local pixels, while the pixels with longer distances have weaker correlation. Therefore, each neuron does not actually need to perceive the global image, it only needs to perceive the locality, and then the local information is integrated at a higher level to obtain the global information. The idea of network connectivity is also inspired by the visual system structure in biology. Neurons in the visual cortex receive information locally (that is, these neurons

respond to stimuli in only certain areas).



But in fact, there are still too many parameters, so start the second-level artifact, that is, weight sharing. In the above local connection, each neuron corresponds to 100 parameters, a total of 1,000,000 neurons. If the 100 parameters of these 1,000,000 neurons are all equal, then the number of parameters becomes 100.

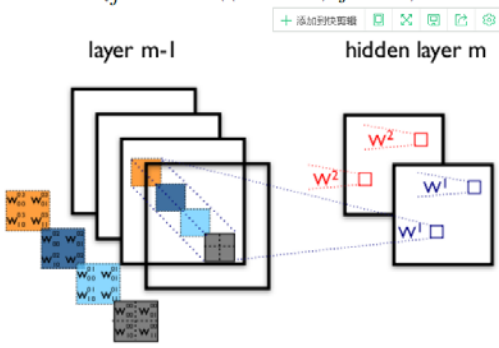
When there are only 100 parameters mentioned above, it indicates that there is only one 10 * 10 convolution kernel. Obviously, feature extraction is not sufficient. We can add multiple convolution kernels, such as 32 convolution kernels, and can learn 32 feature. When there are multiple convolution kernels, as shown below:



Above the right, different colors indicate different convolution kernels. Each convolution kernel generates an image as another image. For example, two convolution kernels can generate two images, and these two images can be regarded as different channels of an image. As shown in the figure below, there is a small error in the figure below, that is, w1 is changed to w0, w2 is changed to w1. They are still referred to hereinafter as w1 and w2.

The following figure shows the convolution operation on four channels. There are two convolution kernels to generate two channels. It should be noted that each channel on the four channels corresponds to a convolution kernel. First, ignore w2 and only look at w1. Then the value at a certain position (i, j) of w1 is determined by the four channels The convolution results at i, j) are added and then the activation function value is obtained.

$$h_{ij}^{k} = \tanh((W^{k} * x)_{ij} + b_{k})$$



Therefore, in the process of convolving 4 channels to obtain 2 channels, the number of parameters is 4 × 2 × 2 × 2, where 4 means 4 channels, the first 2 means 2 channels are generated, and finally 2 × 2 represents the convolution kernel size.

After obtaining the features through convolution, the next step is to use these features for classification. In theory, one can use all the extracted features to train a classifier, such as a softmax classifier, but doing so faces a computational challenge. For example: For a 96X96 pixel image, assuming we have learned 400 features defined on 8X8 input, each feature and image convolution will get a (96 − 8 + 1) × (96 − 8 + 1) = 7921-dimensional convolutional features. Since there are 400 features, each example will get a 7921 × 400 = 3,168,400-dimensional convolutional feature vector. Learning a classifier with more than 3 million feature inputs is inconvenient and prone to over-fitting.

To solve this problem, in order to describe large images, a very natural idea is to aggregate statistics of features at different positions. For example, one can calculate the average (or maximum) value of a specific feature on an area of the image. Not only do these summary statistical features have a much lower dimension (compared to using all the extracted features), they also improve the results (not easy to overfit). This aggregation operation is called pooling (sometimes called average pooling or maximum pooling (depending on the method of computing pooling).

There are two forms of subsampling, one is mean-pooling, and the other is max-pooling. The two kinds of subsampling are regarded as a special convolution process, as shown in the figure below:

(1) Each weight of the convolution kernel of the mean sub-sampling is 0.25, and the step size of the convolution kernel sliding on the original inputX is 2. The effect of mean subsampling is equivalent to reducing the original image blur to a quarter.

(2) In the convolution kernel of the maximum sub-sampling, only one of the weight values is 1, and the rest are 0. The position of 1 in the convolution kernel corresponds to the position where the value of inputX covered by the convolution kernel is the largest. The sliding step of the convolution kernel on the original inputX is 2. The effect of maximum subsampling is to reduce the original image to a quarter, and retain the strongest input in each 2 * 2 region.

The convolution of the forward process is a typical valid convolution process, that is, the convolution kernel kernalW is overlaid on the input map inputX, the corresponding positions are multiplied and summed to obtain a value and assigned

to the corresponding position of the output map OutputY. Each time the convolution kernel moves a position on inputX, it overlaps from left to right from top to bottom and covers it again to get the output matrix outputY (as shown in Figure 4.1 and Figure 4.3). If the input map inputX of the convolution kernel is Mx * Nx and the convolution kernel is Mw * Nw, then the output map Y is (Mx-Mw + 1) * (Nx-Nw + 1).

During the back propagation of the error signal, the error signal of each neuron in the tail classifier is first obtained according to the error back propagation method of the neural network, and then the error signal is propagated by the classifier to the previous feature extractor. The error signal is propagated from the feature map of the sub-sampling layer (subFeatureMap) to the feature map of the previous convolution layer (featureMap) through a full convolution process. The convolution here is slightly different from the convolution in the previous section. If the length of the convolution kernel kernalW is a square matrix of Mw * Mw, the error signal matrix Q_err of the subFeatureMap needs to expand Mw-1 rows or columns up and down, and at the same time, the convolution kernel rotates 180 degrees. The error signal matrix P_err of the subFeatureMap is equal to the error matrix Q_err of the featureMap. The convolution kernel W_rot180 is rotated by 180 degrees.

This program uses a sequential model. The first layer uses a conv2d layer, the second layer uses a conv2d layer, then uses 1st fully connected dense, and then uses 2nd fully connected dense. Using Adam optimizer, it can Tell the model how to adjust the weights, how much to adjust, and the ultimate goal is to make the theta the model weights optimal, so that the cost function f (theta) is minimized. Among all the optimization algorithms, the gradient-based algorithm is the most commonly used, that is, first calculate f (theta) to the gradient of theta d (theta), and theta adjusts along the direction of the gradient descent.

The meaning of each parameter in the program has detailed comments in the program.

One of the difficulties encountered in the construction of the program model is the size setting of the convolution kernel, which defines the receptive field of the convolution. In the early convolutional neural networks (such as AlexNet, ZFNet), some large convolutions were used. Kernels (11 × 11 and 7 × 7). Such large convolution kernels will cause a large increase in the amount of calculations, which is not conducive to training deeper models, and the corresponding calculation performance will also decrease. Later convolutional neural networks (VGG, GoogLeNet, etc.) found that by stacking two 3 × 3 convolution kernels, the same perceptual field of view as 5 × 5 convolution kernels can be obtained, and the amount of parameters will be less. In most cases, stacking smaller convolution kernels is more effective than directly using a single larger convolution kernel.

However, this does not mean that larger convolution kernels have no effect. In some fields, larger convolution kernels can still be used. For example, in the field of

natural language processing, because text content does not have a deep abstraction of features like image data, feature extraction in this field often only requires a shallower neural network. When convolutional neural networks are applied in the field of natural language processing, they are usually composed of shallower convolutional layers, but text features sometimes require a wider receptive field to allow the model to combine more features such as phrases and Character), a larger convolution kernel will be a better choice at this time.

To sum up:

The larger the kernel_size, the better. For the CV field, a smaller convolution kernel (3x3) should be adopted to deepen the network structure. For the NLP field, a larger convolution kernel is more effective.

A single 1 × 1 minimal convolution kernel can only be used to separate convolutions and cannot effectively combine the input original features. A large convolution kernel usually combines too many meaningless features and wastes a lot of calculations. Resources.