Stochastic Gradient Descent(SGD)

We can regard Stochastic Gradient Descent as another version of **Gradient Descent(GD)**. We need to know that Gradient Descent algorithm update the parameters in using **all samples**. However, SGD **just randomly choose one sample** to update the parameter. Therefore, when there are a lot of samples in our dataset, if we use GD to update parameters, then the computation cost can be very very very large. However, if we use SGD, the computation cost is very low. Thus, SGD is always faster than GD. Because SGD algorithm has this characteristic, it is widely used to train some machine learning model such as graphical models where the amount of data is usually very large.

Adam

As for the Gradient Descent and Stochastic Gradient Descent algorithm we mentioned above, they have a common characteristic which is that we need to set the learning rate of these algorithms by ourselves and the learning rate is constant as we run the algorithm. Adam algorithm improves this situation. Specifically, this algorithm **allows each independent variable in the objective function to have its own learning rate**. Besides, along with the increase in the number of iterations, **these learning rates will be constantly adjusted.** We find that Adam is more flexible than SGD and GD. It utilized more information coming from dataset. This algorithm is widely used in deep learning, especially tasks such as computer vision and natural language processing, and had very good performance.

Deep Neural Networks(DNN)

A deep learning network usually consists of an input layer, multiple hidden layers, and an output layer. Each layer consists of many nodes. These nodes are very similar to the neurons in the human brain. When the neurons of the human brain are stimulated enough, it releases the signal. The nodes in the neural network are similar, and when these nodes receive enough information, they release the signal. Specifically, a node can combine input data with a set of weights. The product of the input data and the weight will be inputted into the activation function of the node. The activation function can determine whether the signal released by the node should continue to be transmitted in the network, as well as the distance passed, thereby determining how the signal affects the final result of the network. Deep learning networks are widely used in speech recognition, image recognition, and autonomous driving.

CNN(Convolutional Neural Network)

CNN has become part of the most influential innovation in the field of computer vision. A large number of companies are beginning to use deep learning as the core of services. Google uses it for image searches, and Amazon uses it for product recommendations.

CNN use the same weights or filters in all parts of images. CNN can greatly reduce the number of weights between different layers. Suppose each neuron is connected to only N neurons on the next layer. Then the matrix A between those layers has only N independent weights x. In practice, each neuron can correspond to multiple filters to view edges in different directions.

CNN mimics the way humans recognize pictures. By using CNN, the computer can find steep gradients to find where the imagine changes, and then find the edges to understand imagine. It is done by creating a filter. The dot products between a smooth part and a moving filter window will be smooth. But dot products between edge in the image and moving filter will cause a spike. Those dot products are the “convolution” of the filter and image’s pixel.

A classic structure of CNN look likes as following:

$$\text{Input} \rightarrow \text{Conv} \rightarrow \text{ReLU} \rightarrow \text{Conv} \rightarrow \text{ReLU} \rightarrow \text{Pool} \rightarrow \text{ReLU} \rightarrow \text{Conv} \rightarrow \text{ReLU} \rightarrow \text{Pool} \rightarrow \text{Fully Connected}$$

Backpropagation

Backpropagation can be divided into four parts: forward conduction, loss function, backward conduction, and weight update.

In forward conduction, select a training image and let it pass through the entire neural network. In the first training, since all weights or values of filter are randomly initialized, an output that does not bias toward any number is produced. A neural network with such weights cannot make any reasonable classification.

The loss of the first two training pictures will be extremely high. We want the prediction marker to be the same as the training marker. To do that, we want to minimize the amount of loss. If we think of it as a calculus optimization problem, that is, we want to find out which part of the input (the weight in the example) is directly responsible for the loss of the neural network(L). The partial derivatives of L with respect to the weights x should be zero. Backpropagation uses chain rule to compute derivatives quickly.

In general, forward conduction, loss function, backward conduction, and parameter updating are referred to as a learning cycle. For each training picture, the program will repeat a fixed number of periodic processes. Once the parameter updates on the last training sample are completed, the neural network is expected to get enough training so that the weights in the hierarchy are properly adjusted. Finally, we get an accurate prediction.