Lecture Notes - 2018/11/12
Deep Neural Networks

Team 4
Nick Heppert, 김동민, 김병수, 박민준

November 18, 2018

In this lecture we looked over various types of deep neural networks, which are an autoencoder, RNN and CNN.

1 Auto Encoder

General neural networks use labeled data as their training data and try to minimize error. We call it supervised learning in case where both of input and output values are given. On the contrary, learning methods where only input values are given also exist. We call this learning as unsupervised learning. An autoencoder is an unsupervised version of neural network.

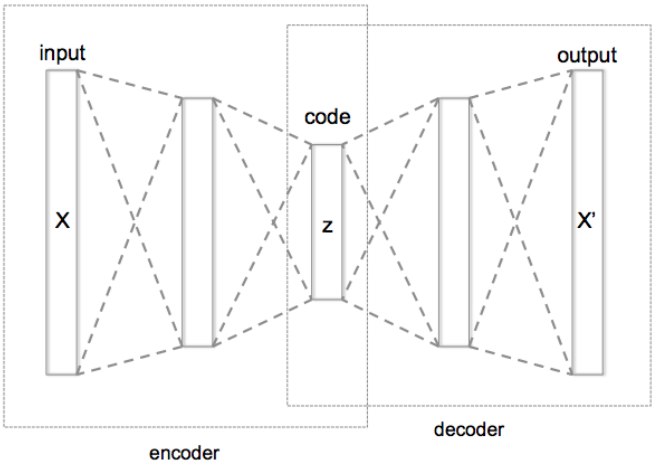

Figure 1: General structure of an autoencoder[1]

As you can see on the figure 1, an autoencoder encodes or compresses input data to certain codes in a hidden layer. Then, uncompress that code into something that closely matches the original data. This forces the autoencoder to engage in dimensionality reduction, for example by learning how to ignore noise.

General procedure to train an autoencoder is as follows.

[1] https://en.wikipedia.org/wiki/Autoencoder

1. Pre-training step; train a sequence of shallow autoencoders greedily one layer at a time.
2. Fine-tuning step1; train the last layer using supervised data.
3. Fine-tuning step2; use back propagation to fine-tune the entire netweok using supervised data.

2 Convolutional Neural Network

In deep learning, a convolutional neural network is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual image.

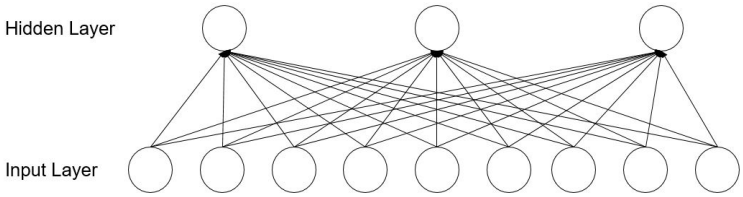

Figure 2: General structure of a neural network

In case of visual image data, there could be too many connections between each layer due to high dimensionality of image data. Too many connections of nodes often not only cause extremely huge computational complexity but also make each node highly affected by adjacent neurons thus learning steps will not go well.

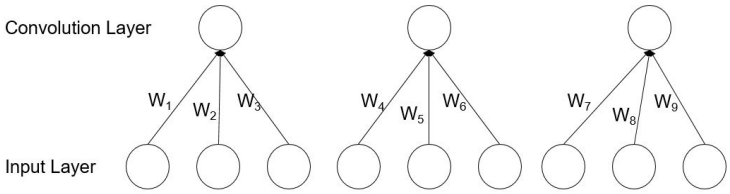

Figure 3: Convolutional Neural Network

By using convolution, we can deal with this kind of problem. we can connect neurons only with adjacent pixels. This scheme is often called "Local connected network" or "Local receptive field" In other words, we make each neurons share their weights. In the given figure, we can set weights as:

$$W_1 = W_4 = W_7 ... \tag{1}$$

W_2 and W_3 follow the same equation. In signal processing, they use this "Sharing weights" method and the terminology is "Convolution."

In a convolutional neural network, we can add a pooling layer which sub-samples output of convolution layers by max-pooling, average-pooling or etc. Suppose input data X_1 is given.

$$X_1 = [0\,1\,0\,0\,0 ...] \tag{2}$$

In this case, W_2 will be selected in the first node of max-pooling layer in figure 4. Then suppose another data set X_2 is given.

$$X_1 = [0\,0\,0\,1\,0 ...] \tag{3}$$

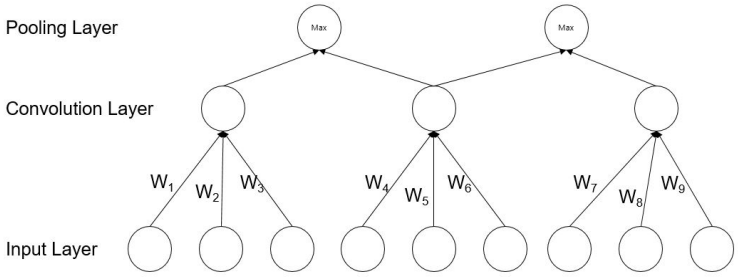

Figure 4: CNN with a pooling layer

In this case, W_2 will be selected again in the first node of max-pooling layer in figure 4. X_1 and X_2 are basically the same data but value of pixel is shifted to the right. In spite of shift, the CNN with a pooling layer produces the same result. That is, output of the max-pooling neurons are invariant to shifts in the inputs and we can say it has translational invariance.

3 Recurrent Neural Network

A recurrent neural network is a class of artificial neural network where connections between nodes form a directed graph along a sequence. Let's suppose we are going to use stock values of recent 10 years. It has continuous sequence along years. Thus, the length of input values could be changed.

$$X = [X_1, X_2, X_3, ..., X_T] \tag{4}$$

If input data is given as above, T changes by data in given examples like stock values. Then how can we deal with it?

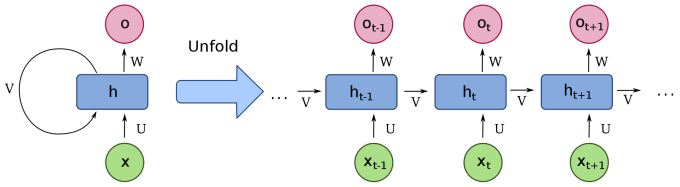

Figure 5: Recurrent Neural Network[2]

RNN is a type of artificial neural network in which the hidden node is connected to the directional edge to form a circulating structure. It is known as a model suitable for processing data that appears sequentially, such as voice and text. Along with Convolutional Neural Networks (CNN), it is a popular algorithm.

As shown in the figure above, RNN's greatest advantage is its ability to create flexible and flexible structures as needed, as it can accept inputs and outputs regardless of sequence length in case of stock value.

[2]https://en.wikipedia.org/wiki/Recurrent_neural_network

4 Appendix

In the following appendix we will provide you with some additional information.

A Long short-term memory

RNNs are known to experience significant degradation of their learning ability as the distance between the related information and the point where the information is used is short, and the gradient of the reverse propagation gradient gradually decreases. This is called the vanishing gradient problem.

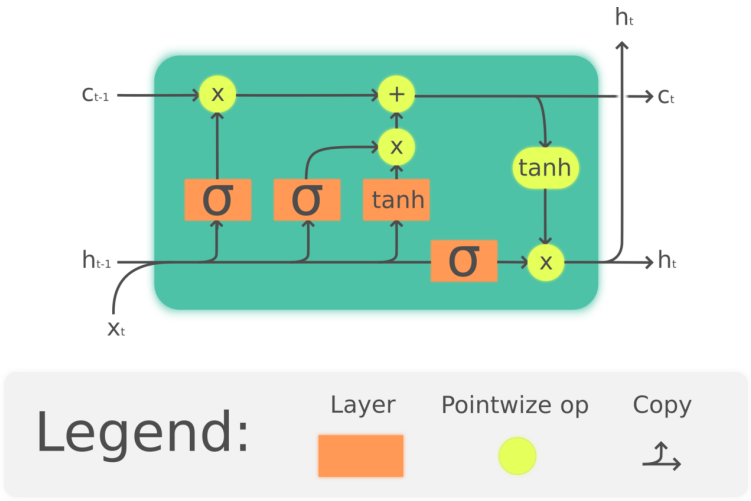

Figure 6: Long short-term memory cell[3]

LSTM is designed to overcome this problem. LSTM is a structure that adds cell-state to the hidden state of RNN. The cell state acts as a kind of conveyor belt. This allows the gradient to propagate fairly well even if the state has been around for quite some time.

Forget gate is a gate for forgetting past information. The value obtained by taking the sigmod with h_{t-1} and x_t is the value that the forget gate exports. Since the output range of the sigmoid function is between 0 and 1, if the value is 0, the previous state information is forgotten and if it is 1, the previous state information is completely memorized.

Input gate is a gate for 'remembering the current information'. It takes a sigmoid by taking h_{t-1} and x_t, taking the hyperbolic tangent with the same input, and then the Hadamard product operation is the output value of the input gate. I personally understood that the range of it is 0 to 1, and the range of gt is -1 to 1, so we understood the intensity and direction respectively.

B Gated Recurrent Unit

GRU is a cell structure that reduces computational complexity while maintaining the advantages of LSTM. GRU is similar to LSTM in that it overcomes the Gradient Vanishing / Explosion problem, but it omits the gate part. GRUs are divided into two groups: update gate and reset gate.

[3]https://en.wikipedia.org/wiki/Long_shortterm_memory
[4]https://en.wikipedia.org/wiki/Gated_recurrent_unit

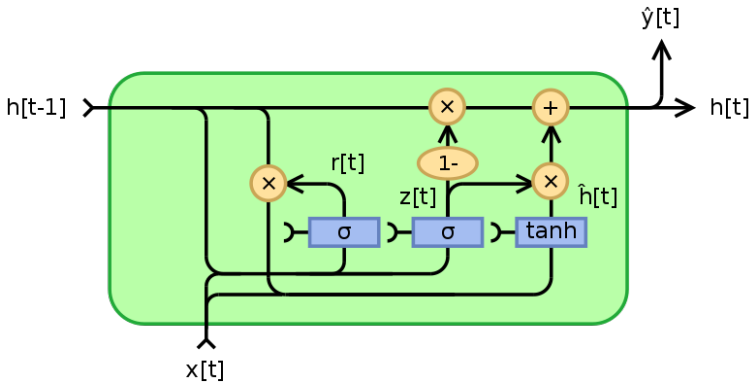

Figure 7: Gated Recurrent Unit cell[4]