Contents

Similarity measure in neural networks

Some comments

This is just a personal note. It does not represent any company or group. If there are any errors, we can discuss them and welcome corrections. Different views can be discussed, and exchanges are welcome.

1.1 Distance and similarity

I think the concept of similarity is some what the core of machine learning. The basic premise is that **similar things should be close, and things that are close should be similar.** The best example is the K-nearest-neighbor algorithm. If I am going to explain this algorithm in one sentence, I would say: **If a person's best 10 friends often go to nightclubs, then this person probably often goes to nightclubs.** Let's consider three individuals:

Person A is from Wuhan, with a height of 179cm, weight of 70kg, aged 25. They enjoy playing basketball, work as a programmer, prefer using Taobao for shopping, and commute by bicycle. **Person B**, also from Wuhan, is 180cm tall, weighs 70kg, aged 26. They share the same interests as Person A: playing basketball, working as a programmer, preferring Taobao for shopping, and commuting by bicycle. **Person C**, from Shanghai, stands at 185cm, weighs 80kg, and is aged 54. Their hobbies include keeping up with the news, they work as an editor, prefer using JD (Jingdong) for shopping, and commute by car.

Further, we can represent these individuals' characteristics numerically. For instance, converting hobbies to numbers—playing basketball is 1, keeping up with the news is 2; converting occupations to numbers—programmer is 1, editor is 2; converting shopping platforms to numbers—Taobao is 1, JD is 2; and commuting methods to numbers—biking is 1, driving is 2. Thus, we can represent these individuals' features in a tabular form:

height	weight	age	hobby	occupation	shopping	commute
179	70	25	1	1	1	1
180	70	26	1	1	1	1
185	80	54	2	2	2	2

Table 1.1: Features

This way, we've transformed people's characteristics into numerical representations, which is called **feature engineering**. In the real world, Person A and Person B should get along well because they share many commonalities. This means that if A likes something, B will probably like it too. **Therefore, a recommendation algorithm can recommend things that A likes to B, and vice versa.** Isn't sales volume determined in this way? If a person likes something, and their friends also like it, then the sales volume of that item will increase.

To make it simple, similarity measure in mathematics is a distance measure. Different data types have different distance calculation methods. For example, cosine distance is often used to compare the similarity of two pieces of text. Feature engineering is particularly important for traditional machine learning methods. This step is mainly operated manually. If feature engineering is not done well, for example, if A and B are similar in reality, but the distance between vector A and

vector B is very far, even farther than the distance between vector A and vector C, then the machine learning effect will be very poor because of low representation quality.

1.2 Similarity in Binary Classification

1.2.1 Linear regression

Regression can be simply understood as a task of finding relationships, seeking the relationship between input and output. You have obersations $x_1, x_2, x_3, ..., x_n$, and corresponding labels $y_1, y_2, y_3, ..., y_n$. and you try to find a function $f(x)$, such that $f(x_i) \approx y_i$. Some relationships between input and output are linear. If you use a linear function to fit this relationship, the effect will be very good, otherwise, the effect will be poor.

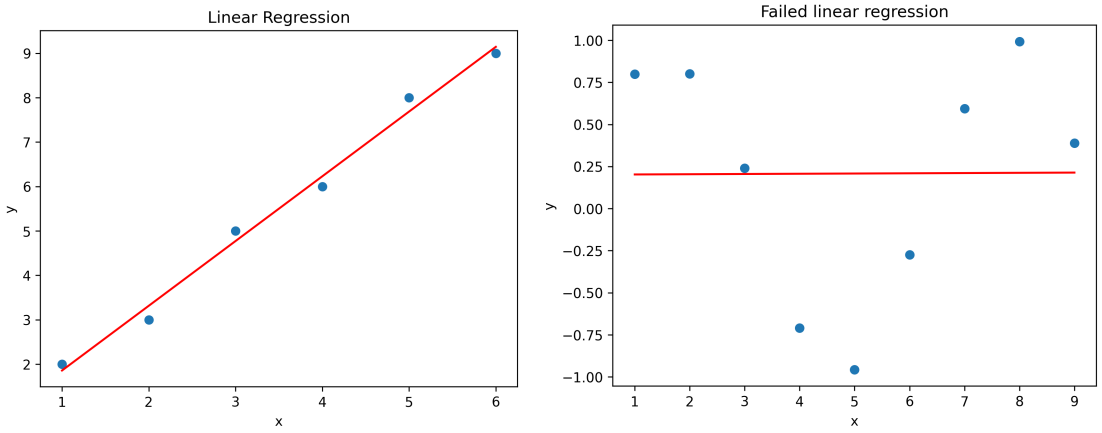

Figure 1.1: Linear regression

Usually, in the regression task, we define the difference between y and $\hat{y}$ as the loss function, and then find the best $f(x)$ by minimizing the loss function.

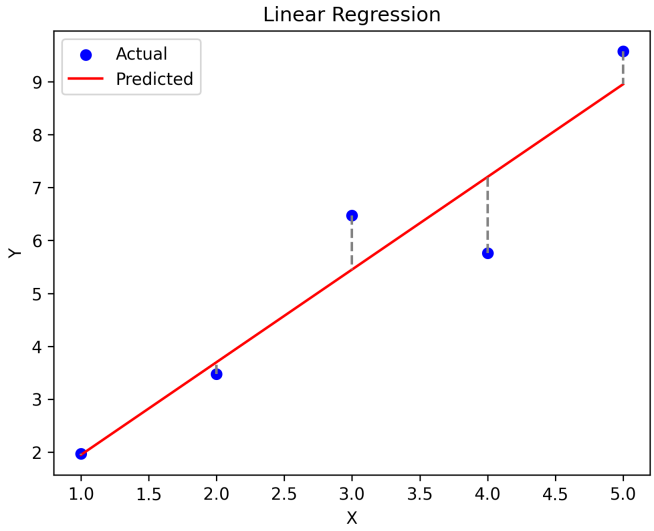

Figure 1.2: Loss is the summation of the difference between y and predicted y

why the metric is not something like Figure 1.3 instead of Figure 1.2? why the loss function in linear regression is represented as a vertical line instead of the distance between the point and the line?

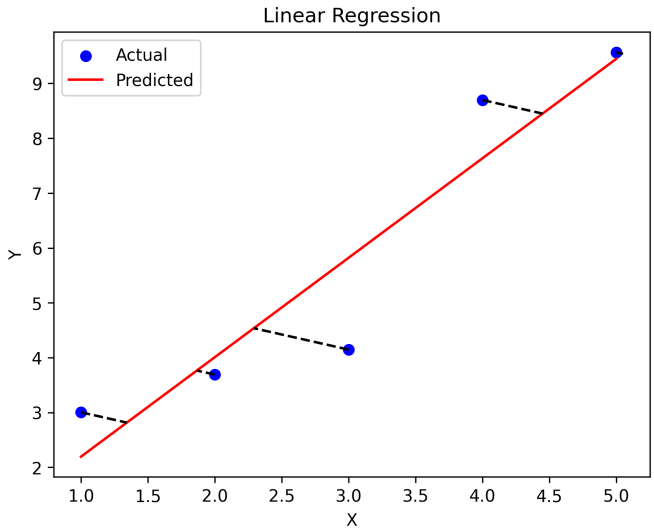

Figure 1.3: Distance: vertical and horizontal

my answer is we don't need both the horizontal and vertical distances, we only need the vertical distance. The vertical distance is the shift in the y-axis, this is what we care about.

1.2.2 Logistic regression

If we divide the points in linear regression into two categories, one above the line and one below the line, then this problem becomes a binary classification problem. The line can be represented as $y = w_1 x + w_0 = 0$, and the points above the line can be represented as $y = w_1 x + w_0 > 0$, and the points below the line can be represented as $y = w_1 x + w_0 < 0$.

$$y = \begin{cases} Blue & \text{if } w_1 x + w_0 > 0 \\ Yellow & \text{if } w_1 x + w_0 < 0 \end{cases} \tag{1.1}$$

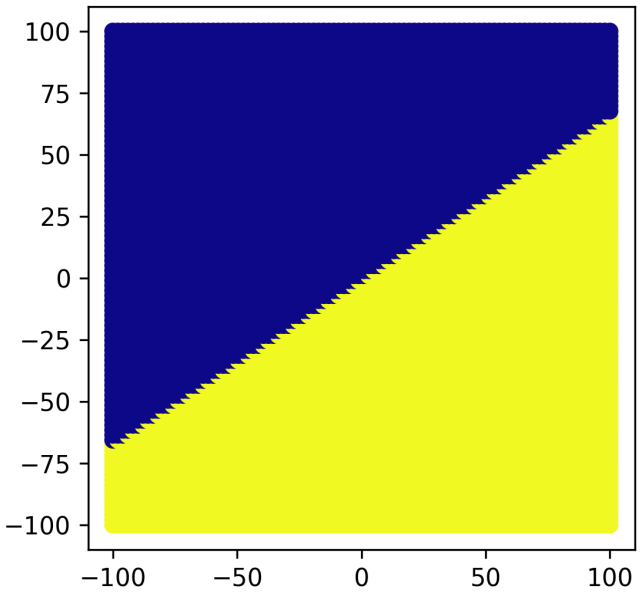

Figure 1.4: Binary classification

Now we have a problem here: from Figure 1.4, we don't know how far the point is from the line. We want to distinguish between the points that are close to the line and the points that are far from the line.

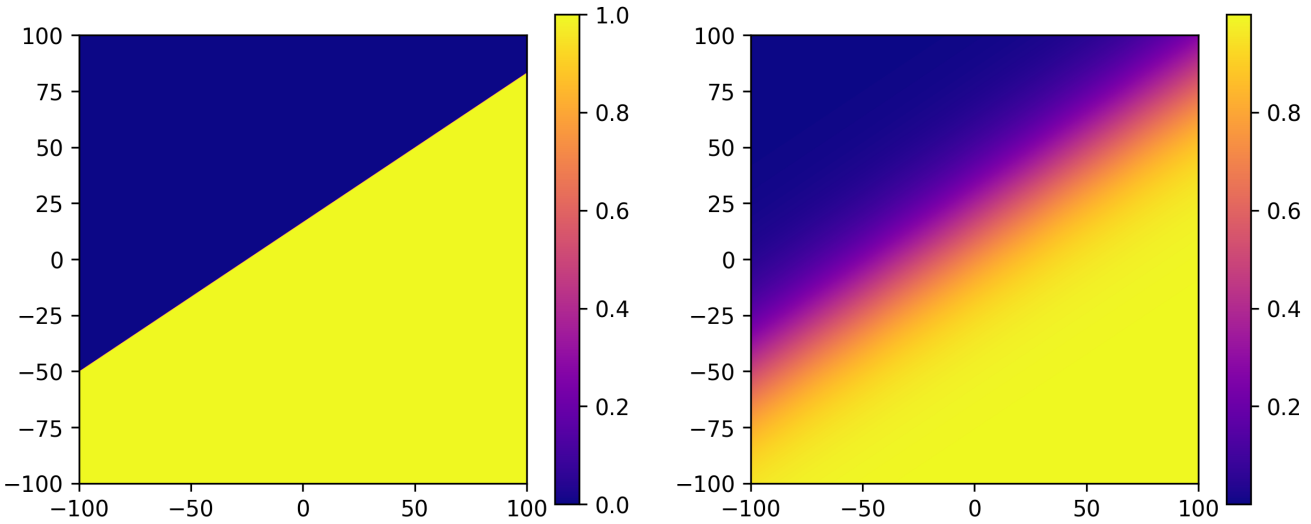

Figure 1.5: Binary classification

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{1.2}$$

To solve this problem, we can take the result of linear regression and convert it into a value between 0 and 1 using a sigmoid function. This value can be understood as the distance of the point from the line. The farther the point is from the line, the closer this value is to 0 or 1, and the closer the point is to the line, the closer this value is to 0.5.

Now let's look at an actual example.

Suppose we have already converted the features of cats (ear shape, nose size) and dogs (ear shape, nose size) into 2-dimensional vectors. Our goal is to distinguish between cats and dogs.

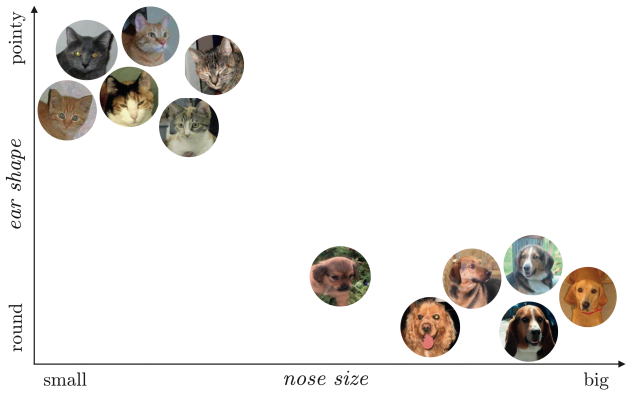

Figure 1.6: Cat and Dog

Image source: BOOK machine learning refined

We need to find a line that can separate cats and dogs. This line is our decision boundary.

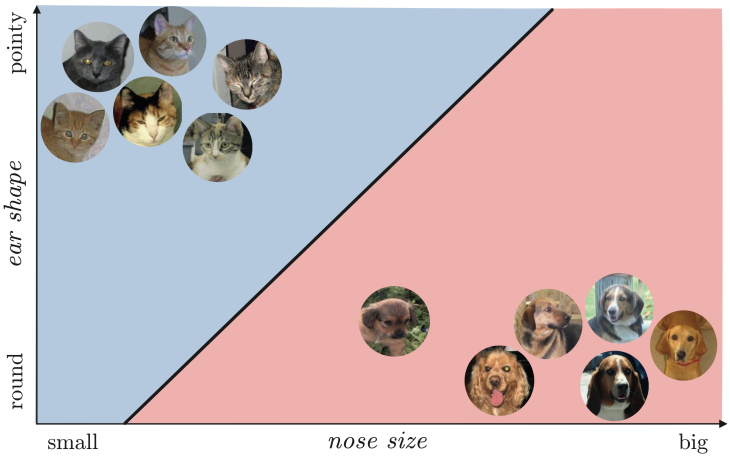

Figure 1.7: Decision boundary

Image source: BOOK machine learning refined

This sentence may be a bit convoluted, but I think you will understand. Some cats are more like cats than other cats. For example, some women have typical "female" facial, some men have typical "male" facial features, but some people are more androgynous. The animals near the decision boundary may be androgynous animals. They have both cat-like and dog-like features. The farther away from the decision boundary, the more obvious their features are.

It is worth mentioning that the output of the sigmoid is a value between 0 and 1, such as 0.3. This means that perhaps 30% of the features are cat-like and 70% are dog-like. This indicates that this may be a dog.Plus, the decision boundary is not necessarily a straight line. It can be a curve, a circle, or any shape.

1.2.3 Some other Classification methods

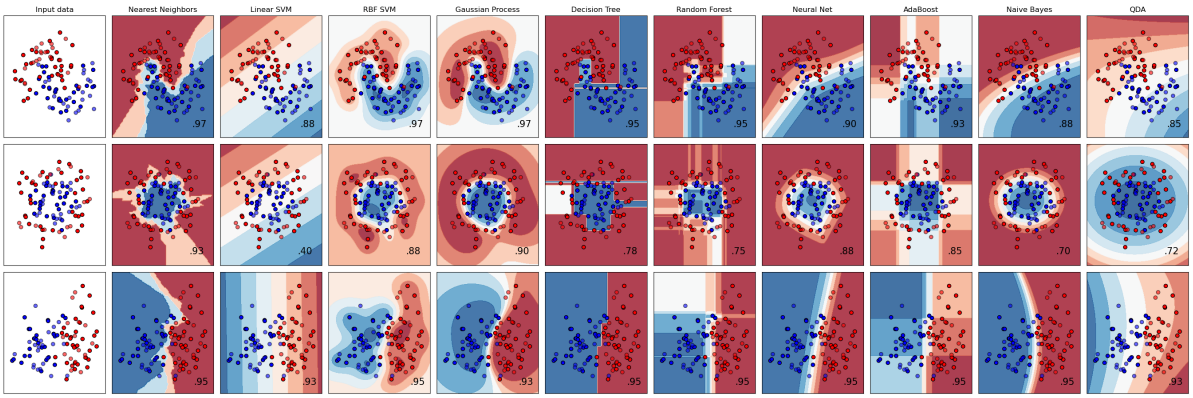

Figure 1.8: Decision boundary by different classifiers

Image source: scikit-learn classifier comparison

In traditional machine learning methods, there are many classifiers, such as SVM, KNN, Decision Tree, Random Forest, etc. The decision boundaries that these classifiers can create are different, and the classification performance depends highly on the distribution of the data. You can think of these classifiers as **function approximators**. Their goal is to find a suitable boundary that can separate data of different classes. However, the ability of these function approximators is limited. They are

only suitable for specific data, meaning they are **not universal**. The design of these traditional machine learning algorithms requires some mathematical foundation, such as convex optimization, probability theory, linear algebra, etc. I think the ideas behind these algorithms are worth learning.

1.3 Similarity in Covlutional Neural Networks(CNN)

CNN includes Convolutional layer, Pooling layer, Fully connected layer. The Convolutional layer extracts features, the Pooling layer reduces dimensionality, and the Fully connected layer classifies. I only plan to talk about the Convolutional layer here, which I think is the most important part of CNN.

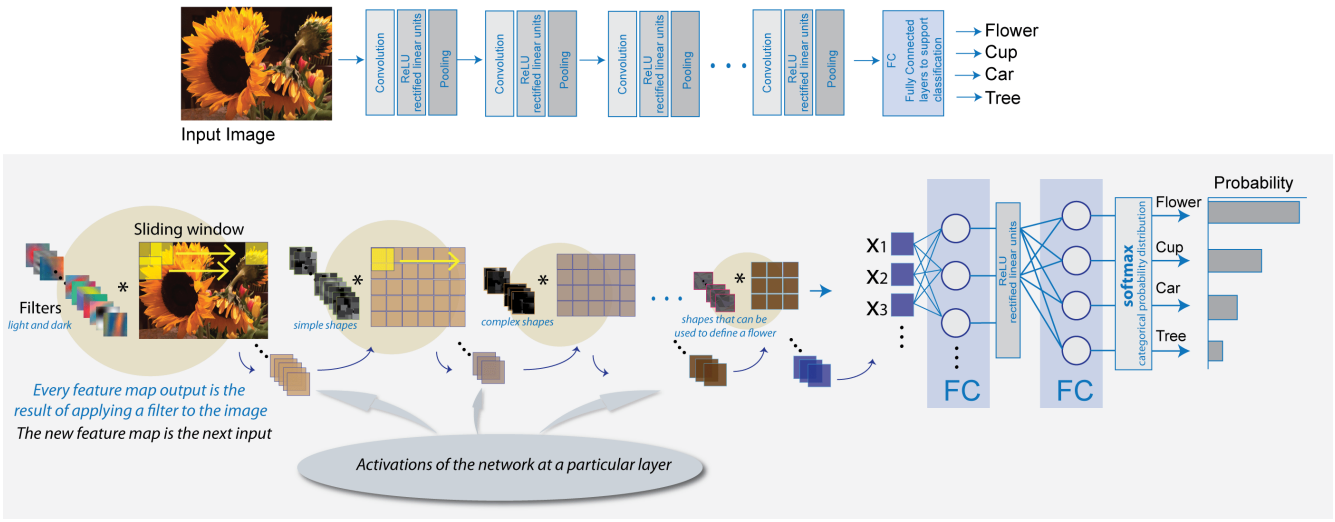

Figure 1.9: CNN

Image source: MathWorks introduction-to-convolutional-neural-networks

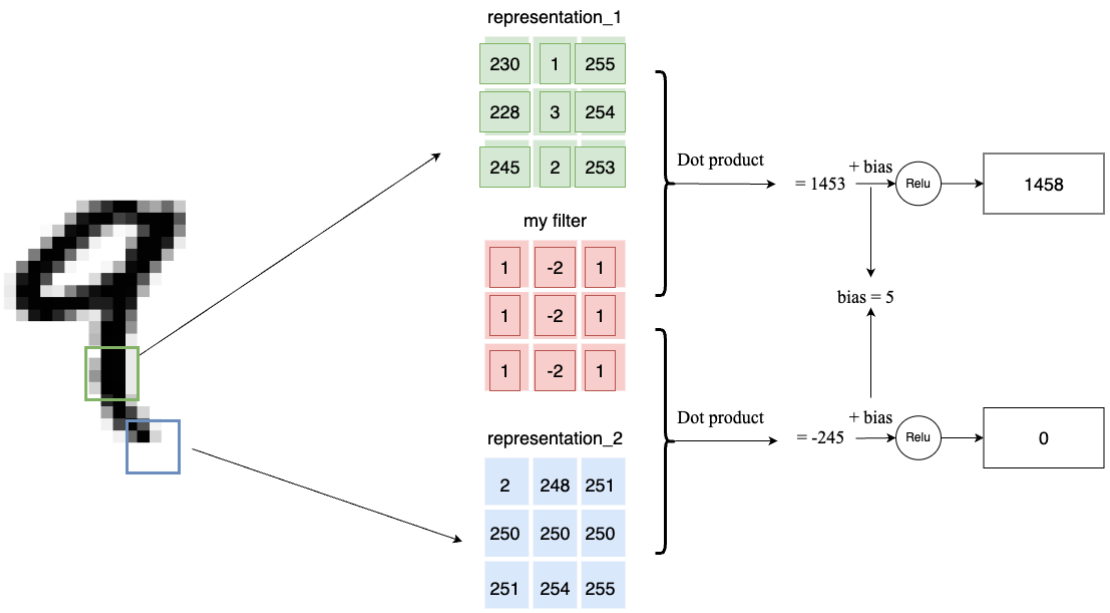

Figure 1.10: Dot product and similarity

The process of the Convolutional layer is to calculate the dot product between the filter and the image representation, then add the bias, and then pass through an activation function, such as ReLU, and finally output a feature map. **The key to this operation is that if the dot product between an image representation and a filter is large, then this image representation and this filter are similar. Conversely, if the numerical value after convolution is very small, then this image representation and this filter are dissimilar.** I think the role of Max pooling further emphasizes this point. The role of Max pooling is to find the largest value in a filter, that is, to select the most similar value.

1.4 Similarity in Deep Neural Networks(DNN)

I think CNN is a special case of DNN. CNN has fewer parameters, but it is essentially a DNN.

For a $28 \times 28 \times 1$ input image, if the number of neurons in the next hidden layer is $28 \times 28 \times 1$ (the image size remains unchanged) and fully connected, then there will be $28 \times 28 \times 1 \times 28 \times 28 \times 1 = 614656$ weight parameters. Using a 3×3 convolution, it can be understood that each neuron in the hidden layer is only connected to 9 neurons in the input layer, and other connections are pruned. **Isn't this a subset of the fully connected layer?**

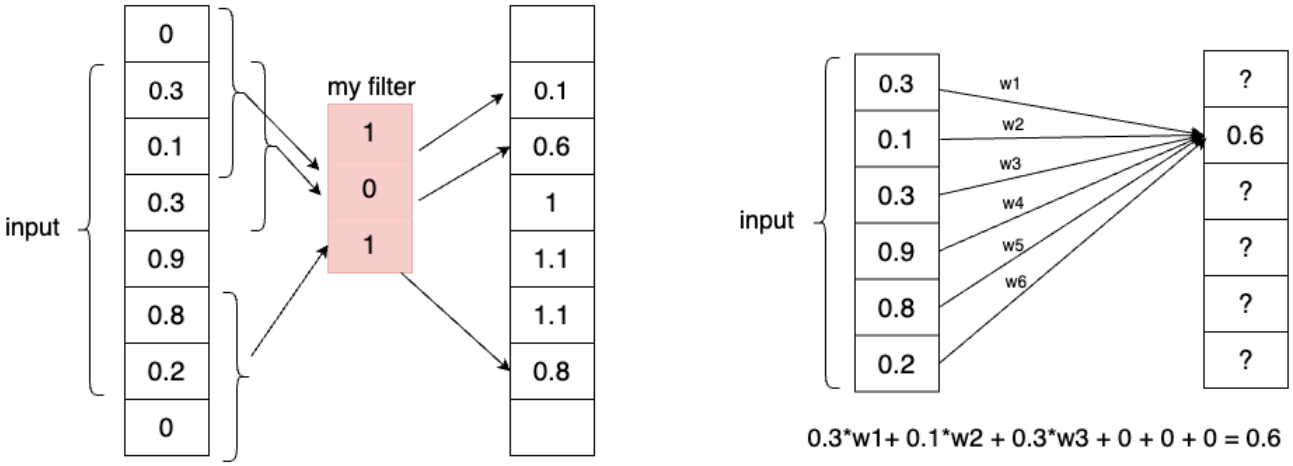

Figure 1.11: CNN and DNN

CNN uses a very clever method to effectively reduce the parameters of DNN. What I want to say is that DNN also has a similarity measure, just like CNN, or in fact, it can achieve the same effect as CNN as long as the number of layers is sufficient.

1.5 Similarity in Attention Mechanism

Attention has 3 components: query, key, and value. Simply put, Q and K do dot product, then let it go through softmax, then multiply by V, and output. How you design K, V, and Q depends on your task. d_k is the dimension of K, $\frac{1}{\sqrt{d_k}}$ is to prevent the value of QK^T from being too large.

$$\text{Attention}(K, V, Q) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1.3}$$

The dot product of Q and K^T is the similarity measure.

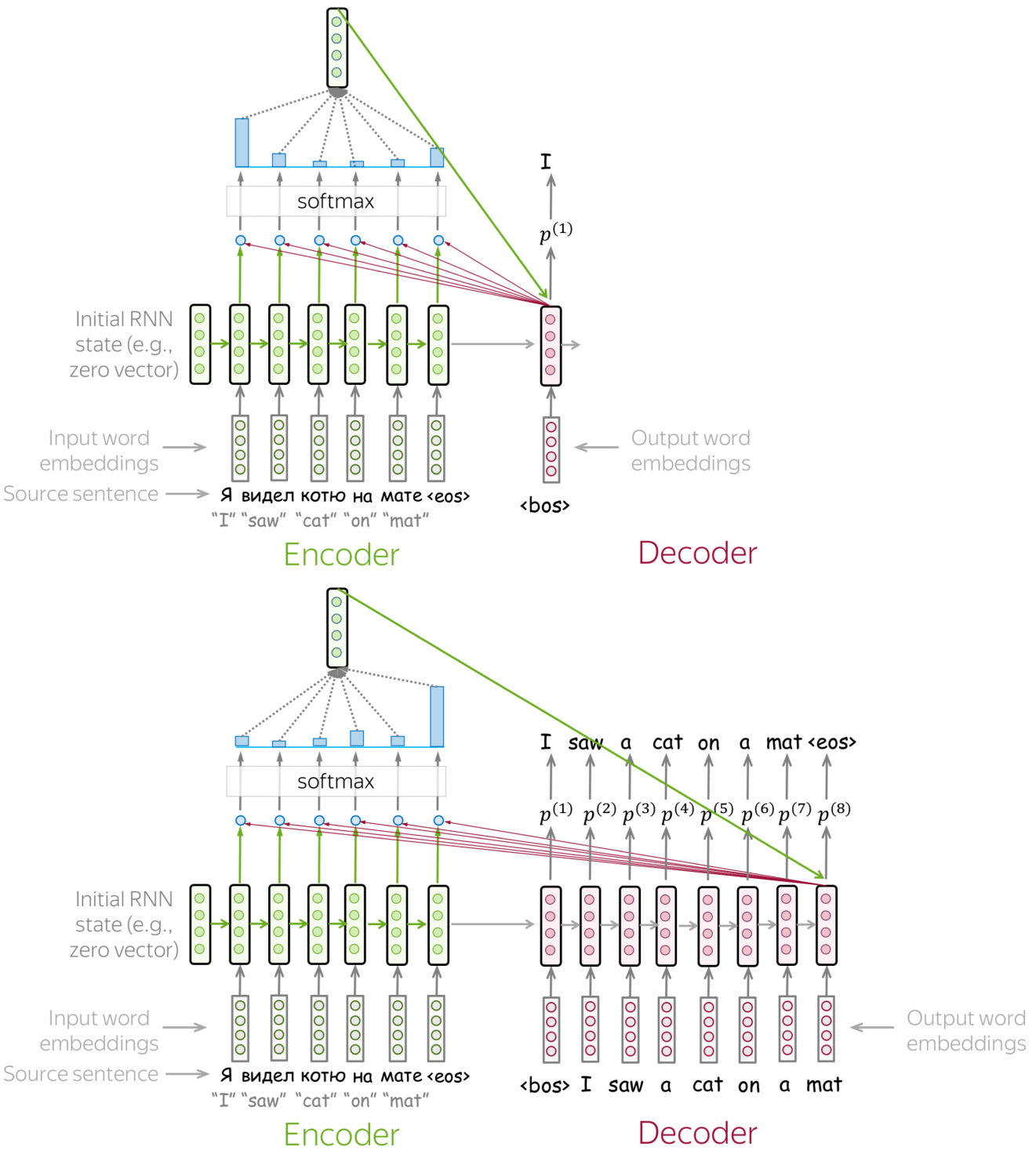

Figure 1.12: Attention for machine translation where Q is the hidden state of the decoder, K and V are the hidden states of the encoder.

Image source: Lena Voita's blog

If you have studied the numerical details of Attention Mechanism, you will find that the essence of attention is a similarity measure. It calculates the similarity, based on which it determines the weight, that is, how to allocate attention.

1.6 Self-attention

Self-attention is a special type of attention, where $Q = K = V$.

1.7 Multi-head Attention and Transformer model

Multi-head attention is a mechanism that allows the model to focus on different parts of the sequence. You can simply think it as a PLUS version of attention mechanism, it has many parallel attention mechanisms, and then concatenate the results.

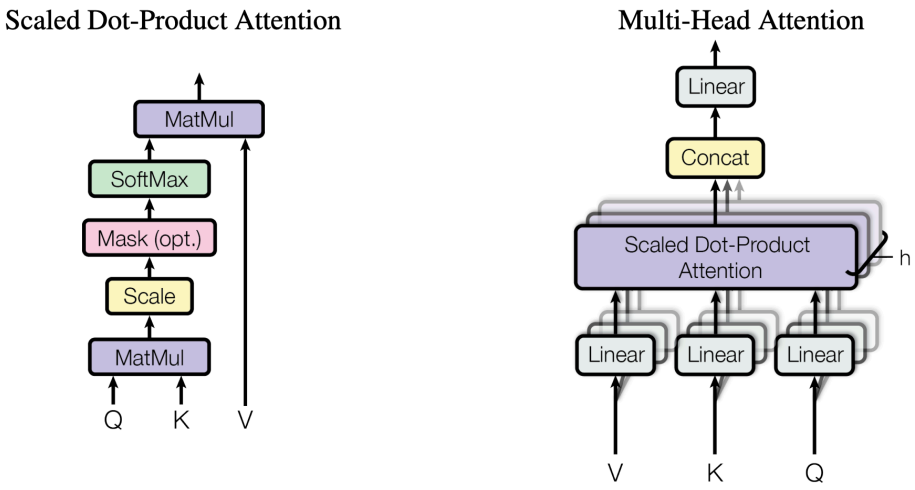

Figure 1.13: Multi-head attention

Image source: PAPER Attention is all you need

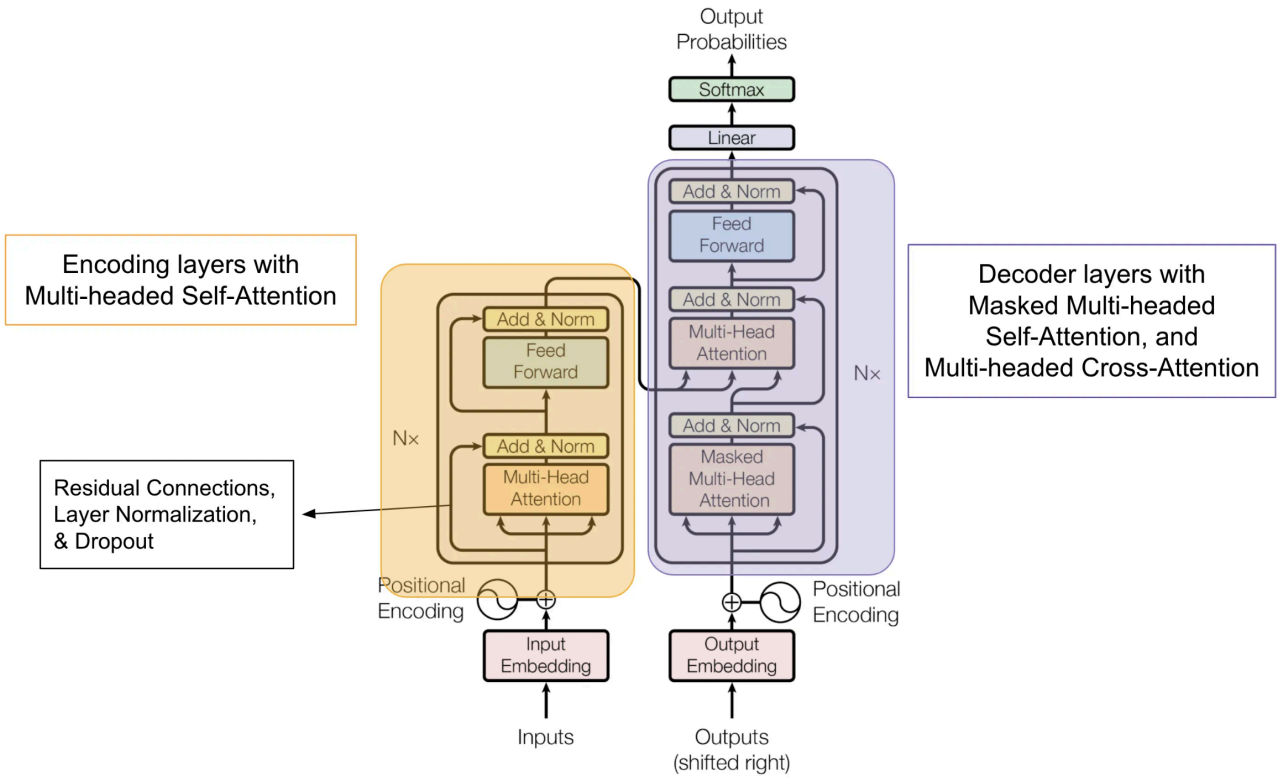

Figure 1.14: Transformer

Image source: PAPER Attention is all you need

I think the core of the Transformer model is the attention mechanism.

1.8 Vision Transformer

In the past few years, CNN was generally used in image processing, while attention was used in text processing. Later, due to the emergence of ChatGPT, transformer-based models achieved great success in the NLP field, so people began to try to apply transformers to image processing. **Essentially, a neural network is a universal function approximator. As long as your network is deep enough and wide enough, you can fit any function.** I am not surprised that transformers can be applied to image processing because whether it is text or image, it needs to be converted into a vector in the end, and then some operations are performed. I have tried to use the vision transformer to do some image processing tasks, and the performance is similar to CNN, which is unexpected. What I expected was that the vision transformer could achieve better results because the vision transformer has more parameters, and the performance should be better. **Maybe their performance is similar because the way they extract features is similar, and the way they calculate similarity is the same, dot product.** Again, it is just a personal note.

1.9 Conclusion

There are many ways to calculate similarity/distance, such as P-Norm similarity, Cosine similarity, Dot product similarity, etc., and you can even define some similarity calculation methods yourself. Whether it is object detection, image classification, text classification, or recommendation systems, similarity calculation is a very important link because if the model cannot grasp the characteristics of things, nothing else can be done. **How to grasp the characteristics of something, that is, to know what is similar to what, and what is dissimilar to what.** In my opinion, CNN, DNN, Attention, and Transformer are using the same similarity measure, which is the dot product. Again, it is just a personal note.