

Yuan An

In last article, I read the guide article of Geitgey [1]. Today, I continue to read the Part 2. The subtitle of this part is Using Machine Learning to generate Super Mario Maker levels (see Fig. 1).



Figure 1. One of the levels the algorithm will generate

Recall the previous part, the author created a simple algorithm that estimate the value of the house based on its attributes. Given data about a house like followings:

Table 1. Attributes of house sales		
Bedrooms	Sq.feet	Neighborhood
3	2000	Hipsteron

The value of the house is estimated by multiplying each of its attributes by a weight, then add all those numbers up to get the house’s value.

The diagram of this procedure are as following Fig. 2:

But this algorithm only works for simple problems where the result has a linear relationship with the input. To make the algorithm more clever, run it multiple times with different weights to capture different edge case (as shown in Fig. 3):

Then combine those four price estimates into on final estimate as following Fig. 4:

Now a ‘Super Answer’ is obtained to model more cases.

Then the author introduced the Neural Network. See in Fig. 5

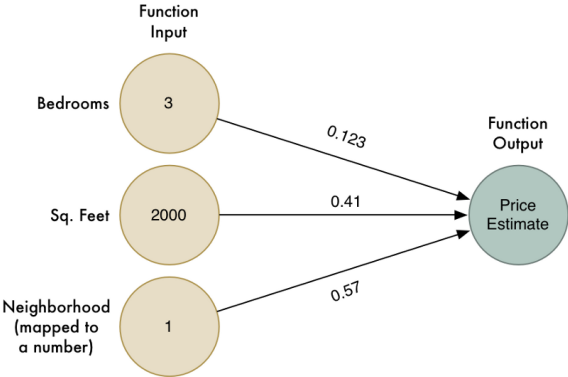


Figure 2. The diagram of estimating procedure

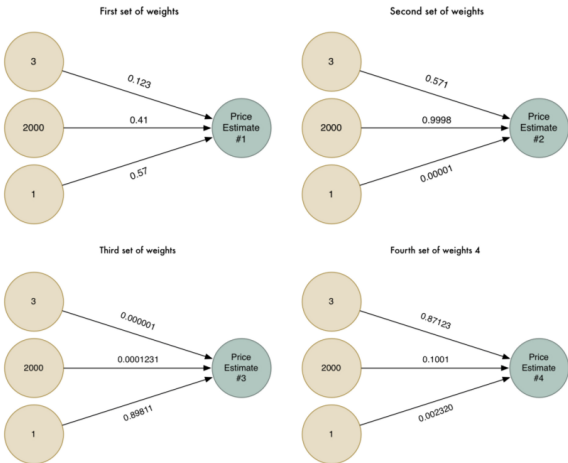


Figure 3. Running four times with different weights

This is a neural network. Each code knows about how to take in a set of inputs, apply weights to them, and calculate an output value. By chaining together lots of these nodes, it can be used to model complex functions. In the neural network, there are important part as following:

- The function that takes in a set of inputs and multiplies them by weights to get an output named neuron
- By chaining lots of simple neuron together, it can model functions that are too complicated to be modeled by one single neuron.



Figure 4. Combine four price into the same algorithm using a different weights set

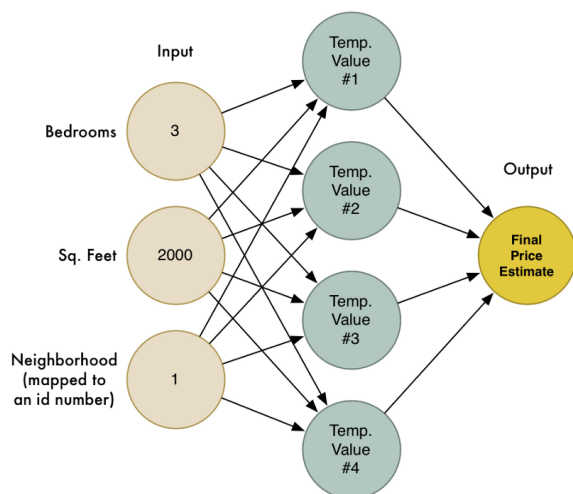


Figure 5. The diagram of the Neural Network

But the neural network made by this method always returns the same answer when given the same inputs. It has no memory, also called stateless algorithm in programming terms. In many conditions, it is needed as we want. But this kind of model can't respond to patterns in data over time.

The author take an example of writing story on computer using keyboard. What computer guess the very first letter the user will type.

It can use the knowledge of English to increase its odds of guessing the right letter. For example, the typer will probably type a letter that is common at the beginning of words. If it looked at stories the typer wrote in the past, it could narrow it down further based on the words typer usually use at the beginning of sto-

ries.

When it comes to guess the next letter the typer will type, this is a much more interesting problem. The author used The Sun also Rises of Ernests Hemingway's novel as an example to introduce the stateful model. The diagram of the stateful model is shown in the Fig. 6

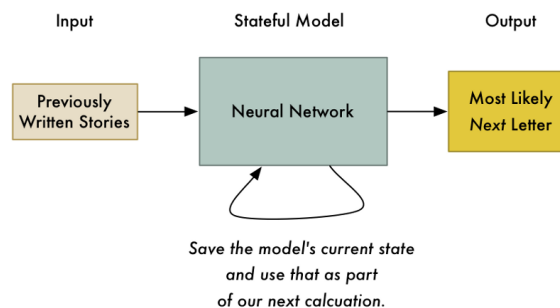


Figure 6. The diagram of the stateful model

For solving the problem, state is add to the model. Each time we can ask our neural network for an answer, and we aslo save a set of our intermediate calculations and re-use them the next times as part of our input. After doing this, the model will adjust its predictions based on the input that it has seen recently. Keeping track of state in the model makes it possible to not just predict the most likely first letter in the story, but to predict the most likely next letter given all previous letters.

The thing talking above is the basic idea of a recurrent neural network. We can update the network each time we use it, and it allows to update its predictions based on what it saw most recently. It can even model patterns over time as long as we give it enough of a memory.

Then the author introduced applications of this model, and used the model created by Andrej Karpathy<sup>1</sup> to generate a story. What's more, the author also created Super Mario's new level using the model.

As machine learning becomes more important in more industries, the difference between a good program and a bad program will be how much data you have to train your models. That's why companies like Google and Facebook need your data so badly. And Google recently released a open source TensorFlow, its software toolkit for building large-scale machine learning applications. It was a pretty big deal that Google gave away such important, capable technology for free.

<sup>1</sup>A deep learning researcher at Stanford and he wrote an excellent introduction to generating text with RNNs.

## References

- [1] A. Geitgey, “Machine learning is fun,” <https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471>.