How to combine categorical and continuous input features for neural network training

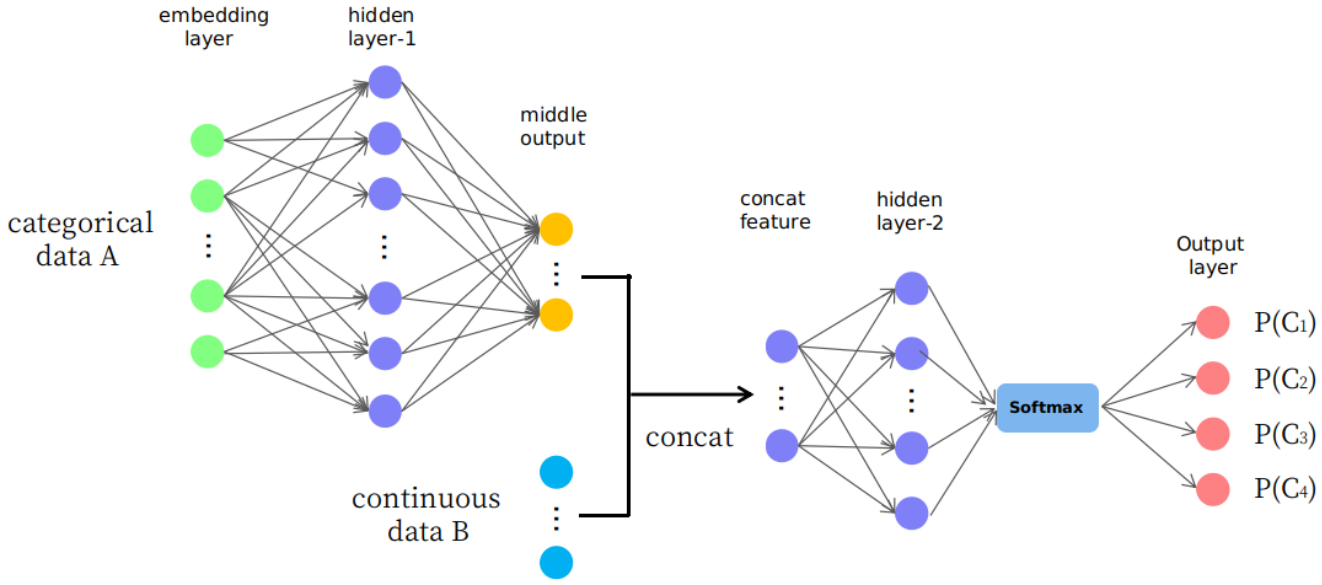
Asked 3 years, 10 months ago Active 1 year, 5 months ago Viewed 9k times

21

14

Suppose we have two kinds of input features, categorical and continuous. The categorical data may be represented as one-hot code A, while the continuous data is just a vector B in N-dimension space. It seems that simply using concat(A, B) is not a good choice because A, B are totally different kinds of data. For example, unlike B, there is no numerical order in A. So my question is how to combine such two kinds of data or is there any conventional method to handle them.

In fact, I propose a naive structure as presented in the picture



As you see, the first few layers are used to change(or map) data A to some middle output in continuous space and it is then being concated with data B which forms a new input feature in continuous space for later layers. I wonder whether it is reasonable or it is just a "trial and error" game. Thank you.

neural-network | feature-selection | categorical-data | feature-construction

Share Improve this question Follow

asked Mar 28 '18 at 8:49

JunjieChen
445 4 8

3 Answers

Active | Oldest | Votes

7

There's three main approaches to solving this:

1. Building two models separately and then training an ensemble algorithm that receives the output of the two models as an input
2. Concating all the data into a single vector/tensor as a preprocessing step and then train a simple single input NN
3. The multiple input NN architecture you proposed

The ensemble approach is the most straight forward option and will yield decent results, however, it will not work as well as the option you proposed as because the ensemble network only receives class probabilities from the two networks as an input and will in

comparison to your approach miss out on more complex relationships between the data types.

The second approach is in theory not that different from your proposed approach, with the difference being that it assumes that the network will on it's own figure out that the input consists of two types of data (as they are both in the same vector/tensor). It will take a lot of training time for the network to learn that and you might get stuck in a local minima before that even happens.

Based on my personal experience, the network you proposed is the best option as and it's likely to have shortest train time and once you get the architecture right you will find it very easy to train and maintain (retrain) the network in production as well as you will only ever have to retrain one single model.

Share Improve this answer Follow

answered Nov 10 '18 at 14:24

Tadej Magajna
171 1 5

1

I had been using the naive structure proposed by you for quite some time by now. In a well framed problem and with enough data, this type of architecture works quite well. However here are a few things which I learnt:

1. Tree based algorithms (RF, XGB) generally performs well with mixed classes unless you have some specific output requirements or loss function which is easier to implement via neural networks.
2. If using neural network is decided, then this architecture performs better compared to other type of string-encoding ways.
3. This approach also works with mixed input time-series data - much better than any classical time series approaches.

The key design would be the concatenation layer and where would you want to put it in the architecture. Also using embedding layers gives you additional benefit of using those learnt embeddings in some other tasks / visualizations.

These type of architecture has been used in Kaggle competitions and is also taught in the Fast.ai course by Prof. Jeremy Howard.

Share Improve this answer Follow

edited Aug 5 '20 at 11:06

Zephyr
997 4 7 19

answered Dec 13 '19 at 12:53

Soumyajit
111 2

0

As @Tadej Magajna already said there are different solutions for that. I would like to add a post about how to implement the third approach in keras:

Combining numerical and text features in (deep) neural networks

Share Improve this answer Follow

answered Nov 18 '19 at 14:38

ixeption

101 2