
Neural Complete

Neural Complete is autocomplete based on a generative LSTM neural network, trained not only *by* python code but also *on* python source code.

Ironically, it is trained on files containing keras imports. The result is a neural network trained to help writing neural network code.

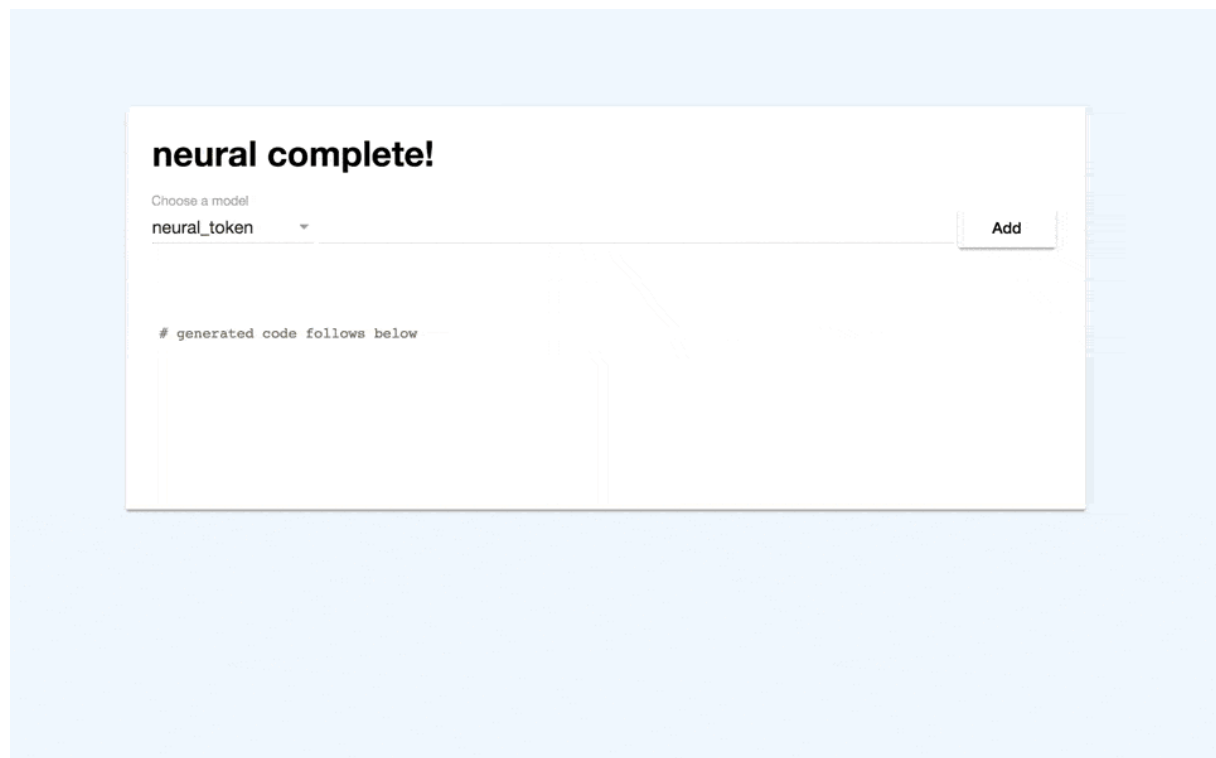
Rather than completing a word, it will suggest finishing a whole line. It uses information from previous lines to make a suggestion.

One could imagine that everyone will have a neural network to automagically complete their personal scripts based on their own neural model :-)

But not yet with this code.

You're encouraged to train on your own data, which should be made easier by using Neural Complete.

Demo



The first time `model` is written, it suggests to create it as a variable (`model = Sequential()`).

The second time `model` is written, it suggests using it instead (`model.add(...)`). It shows that it is able to use the context!

The final line does contain mistakes, but should get more precise with more data and also more context.

Models

There are 2 models included, a character based model and a python token model. The benefit of the char based model is that it can complete at any moment, while the token based model only works with completed tokens (it cannot finish a word). However, the token based model is based on a higher level unit (semantic), and should make more sense most of the time.

The char based model looks back up to 80 characters, while the token based model looks back up to 20 tokens.

It would be very fun to experiment with a future model in which it will use the python AST and take out variable naming out of the equation.

Do It Yourself

Scraping data

Unfortunately the Github API does not allow to search by filename, so I wrote a scraping script to gather python data specifically trained on keras source code. You can change the search query to gather your own data. Do not overdo it as to “annoy” github. You would need a lot more for a reasonable result!

The models have only been trained on 26 scripts.

Backend

Train a model using keras, serve it with flask.

See backend

Frontend

The frontend is a very thin layer communicating with the backend to receive autocomplete suggestions, written in Angular 2. The dist folder has been included so you can easily run it yourself without dependencies.

See frontend

Credits

It uses a lot of the ideas of the standard keras LSTM text generation example.

Whenever using any of this code: please attribute whenever you can.