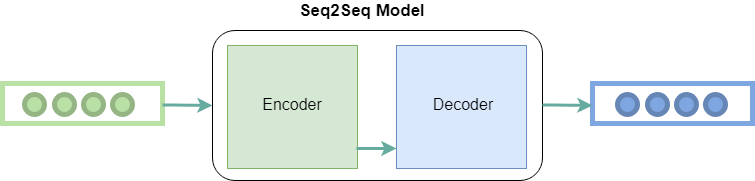
**Seq2seq model for chatbot**

**1 . Introduce Seq2seq model.**

**2 . LSTM.**

**3 . GRU.**

**3 . Attenion mechanism.**



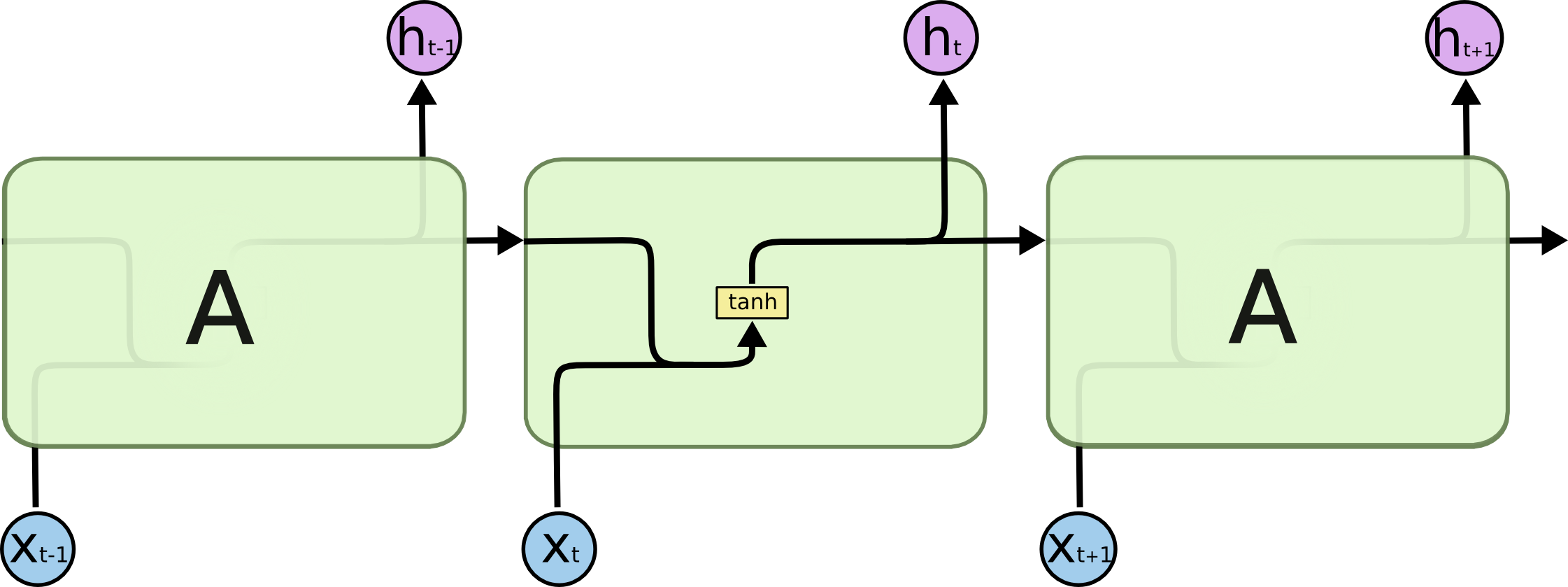
Sequence to Sequence Model (Seq2seq) is a Deep Learning model with the purpose of creating an output sequence from an input sequence whose length of these two sequences may vary. Seq2seq was introduced by the Google team in 2014 in the article Sequence to Sequence with Neural Networks. Although the original purpose of this model was to apply in Machine Translation, but now Seq2seq is also applied in many other systems such as Speech recognition, Text summarization, Image captioning.

Seq2seq includes 2 main parts: Encoder and Decoder.

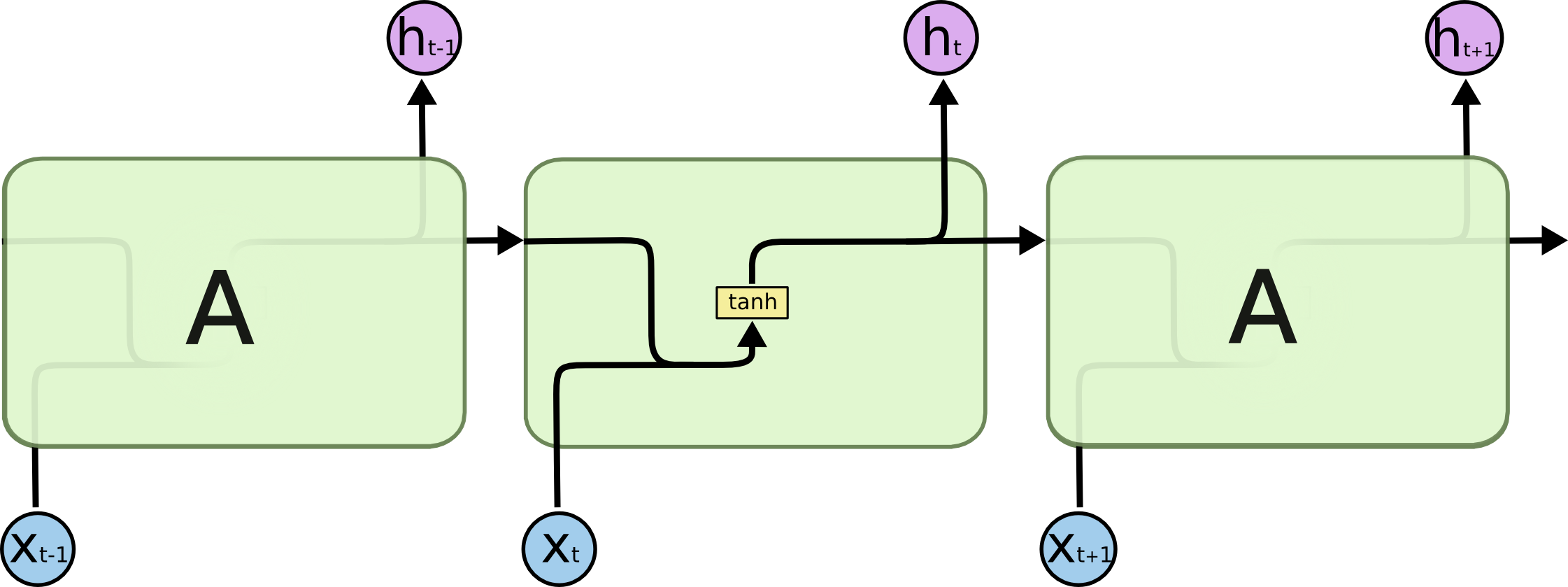
Both components are made up of Neural Networks, in which the Encoder converts input data into an representation with a lower dimension and Decoder is responsible for creating the output sequence from the representation of the input that is generated in the Encoder section.

**LSTM**

* Inencoder phase is responsible for understanding the information of the input data. This understanding means modeling the nature of the input data. For example, in our problem it is the new Vietnamese language. Normally, people will use the regression neural network - RNN to represent information for data in NLP because NLP data is dependent on the chain.
* Long Short Term Memory networks, commonly known as LSTMs - are a special type of RNN, and are capable of learning remote dependencies. LSTM was introduced by Hochreiter & Schmidhuber (1997), and was later improved and popularized by many people in the industry. They work extremely well on many different problems, so they have become popular as they are today.



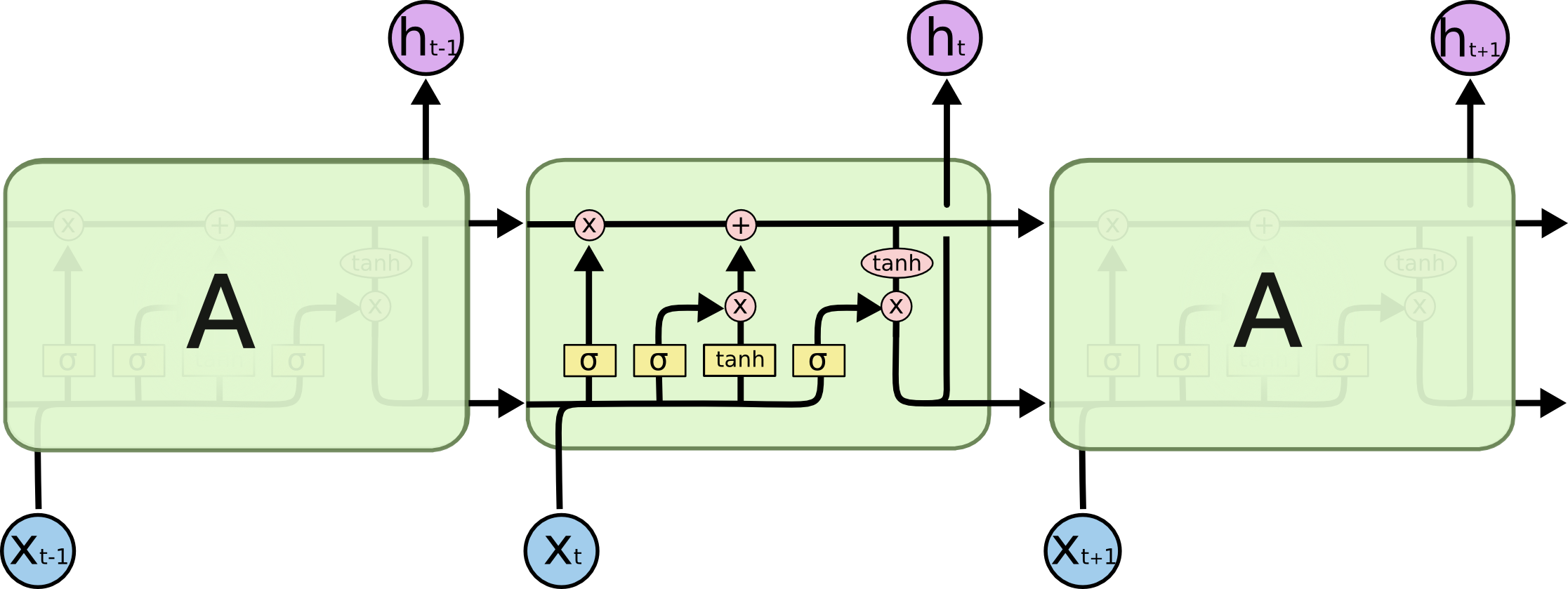
* RNN



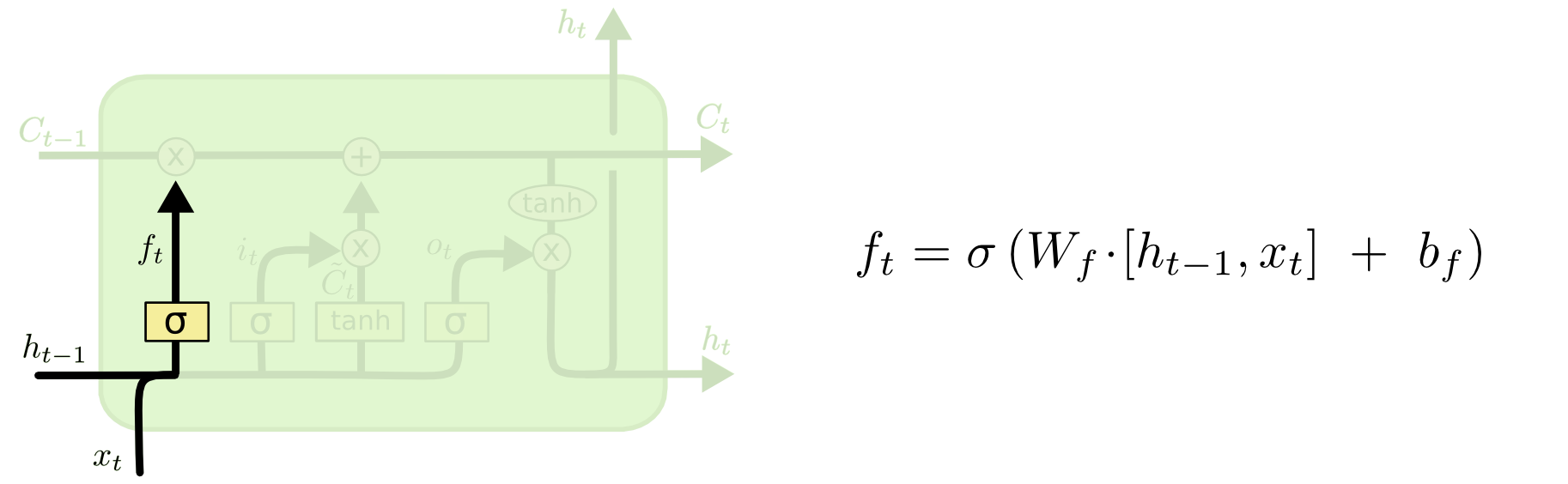




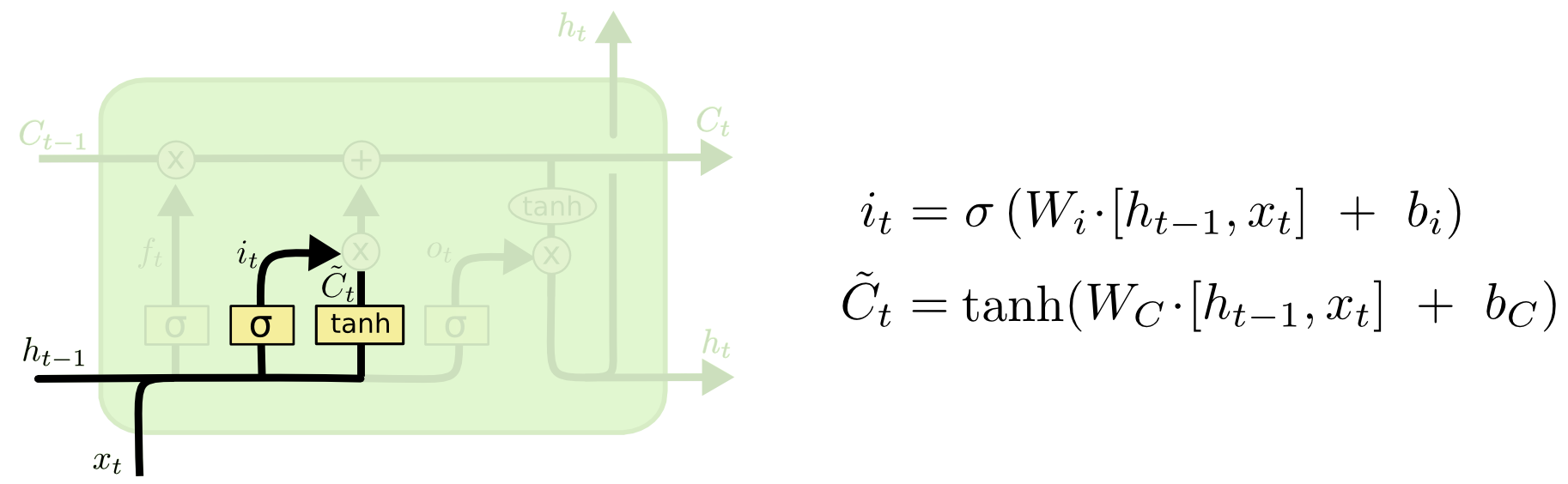
LSTM



The first step of LSTM is to decide which information to remove from the cell state. This decision was made by the sigmoid layer - called the "forget gate layer".

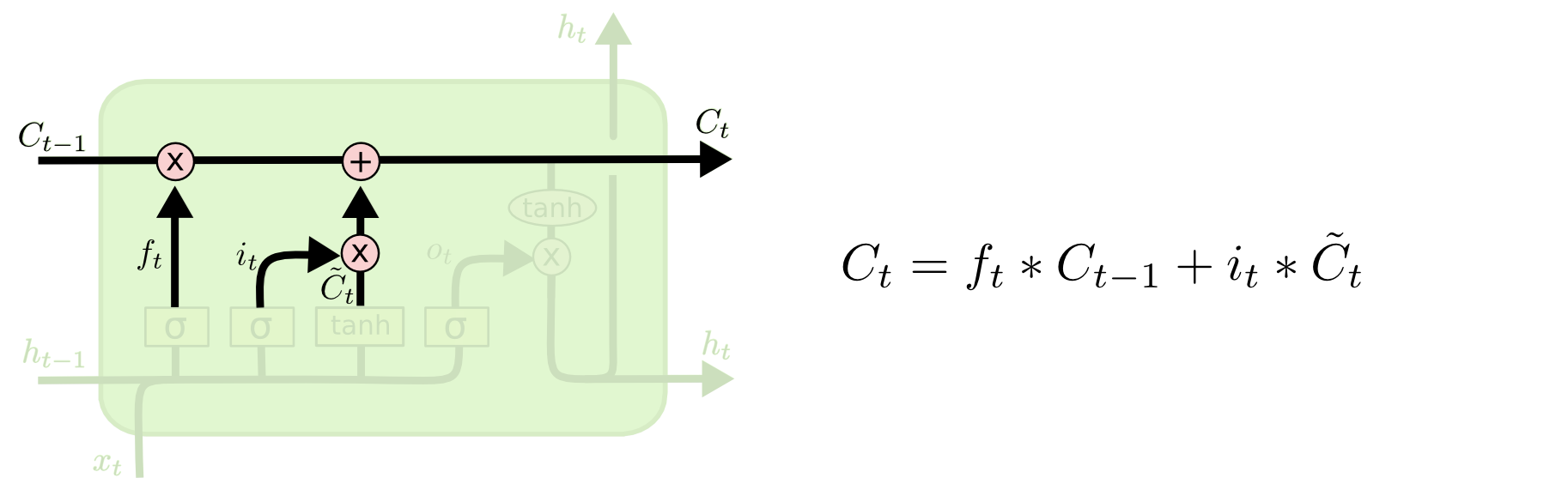


The next step is to decide which new information we will save into the cell state. This consists of 2 parts. The first is to use a sigmoid called the "input gate layer" to determine which values we will update. Next is a fishy floor that creates a vector for new value{C\_t}to add to the status. In the next step, we will combine those two values to create an update for the state.

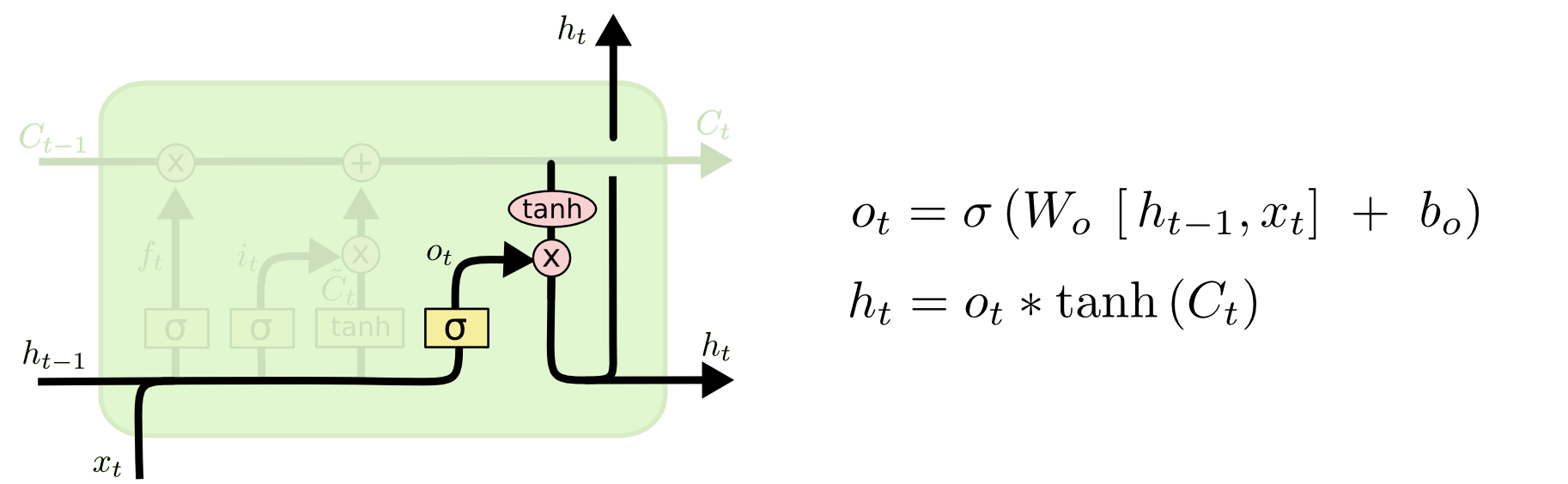


Now it's time to update the old cell state

into a new state



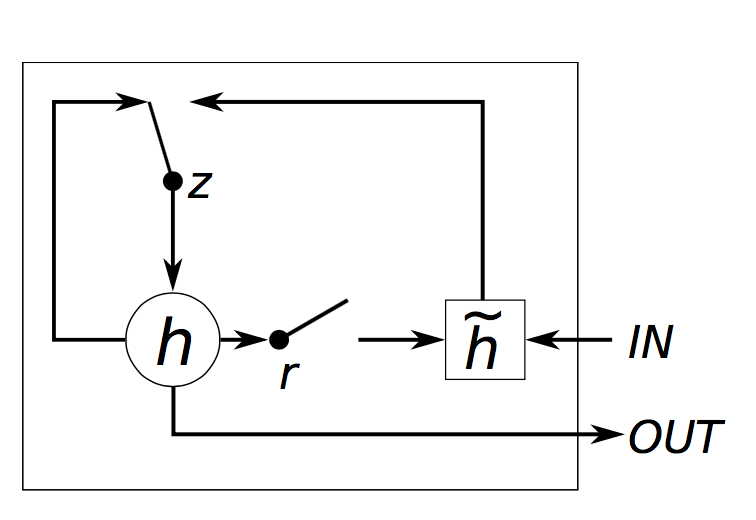
A Finally, we need to decide what we want the output to be. The output value will be based on cell status, but will be further screened. First, we run a sigmoid layer to decide which part of the cell state we want to output. After that, we put it into the state of the cell through a fishy function to get its value to about [-1, 1], and multiply it by the output of the sigmoid port to get the desired output value.



**GRU**

The idea of GRU is quite similar to that of LSTM:

GRU has only 2 ports: r reset port and z update port. The reset port will determine how to combine the current input with the memory first, and the update port will specify how much information about the memory first should be kept.



LSTM and GRU difference in the following ways:

GRU has 2 ports, while LSTM has 3 ports.

GRU has no internal memory Ct

And there is no output port like LSTM.

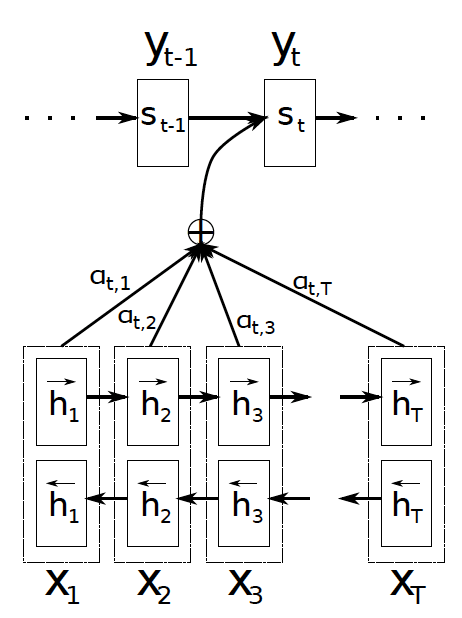
The two input ports and the forgotten ports are combined into the z update port and the r reset port will be applied directly to the pre-hidden state.

GRU does not use a non-linear function to calculate the output as LSTM.

**Attention**

The basic seq2seq model has the disadvantage of requiring the RNN decoder to use all encoded information from the input sequence whether it is long or short. Second, the RNN encoder needs to encode the input sequence into a single vector and has a fixed length.

At the abstract level, the attention technique loosens the condition that the entire input sequence is encoded by a single vector. Instead the words in the input sequence will be encoded by the RNN encoder into a sequence of vectors. Then the RNN decoder applies soft attention technique by taking the weighted sum of the sequence of coding vectors.

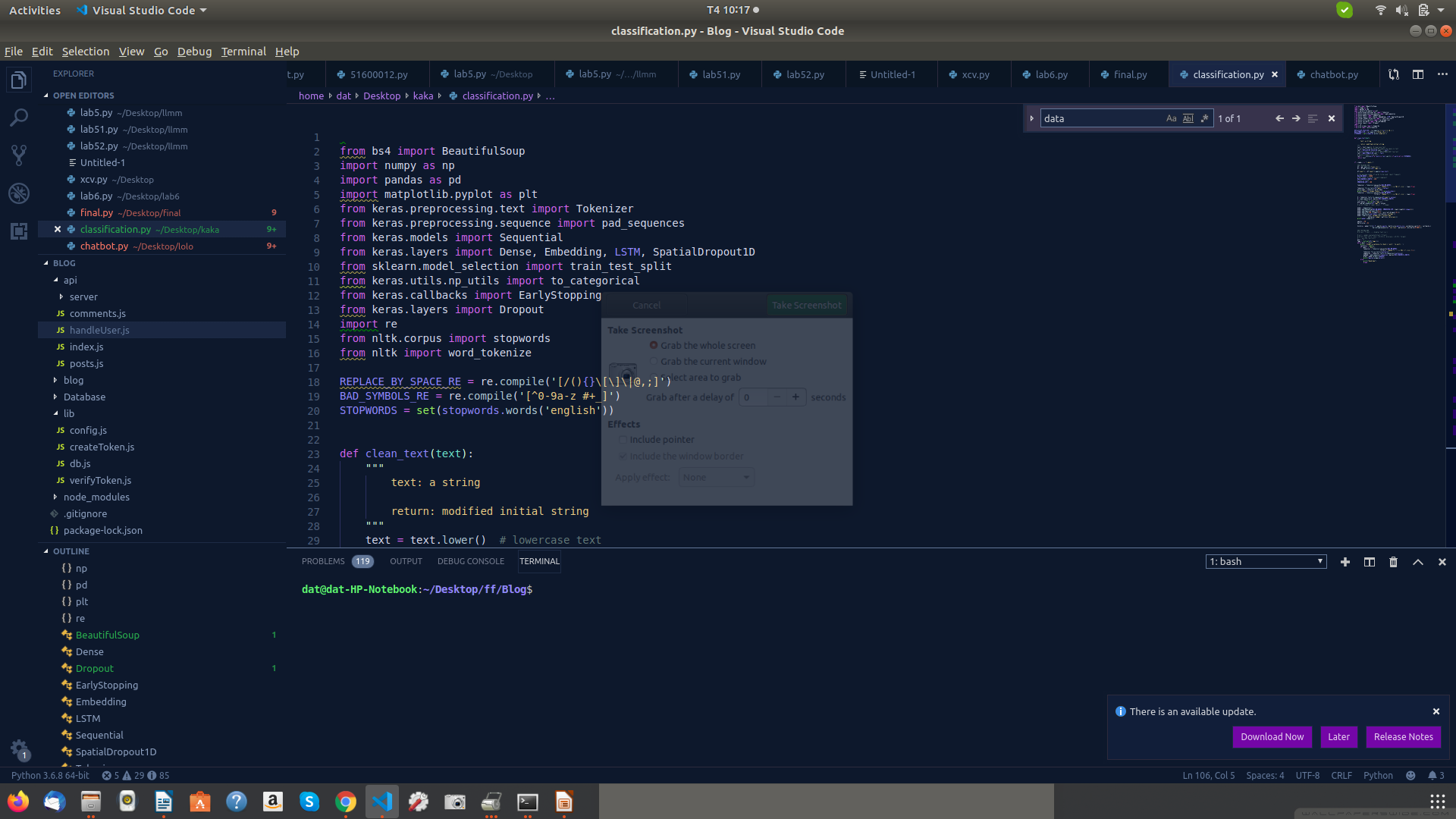


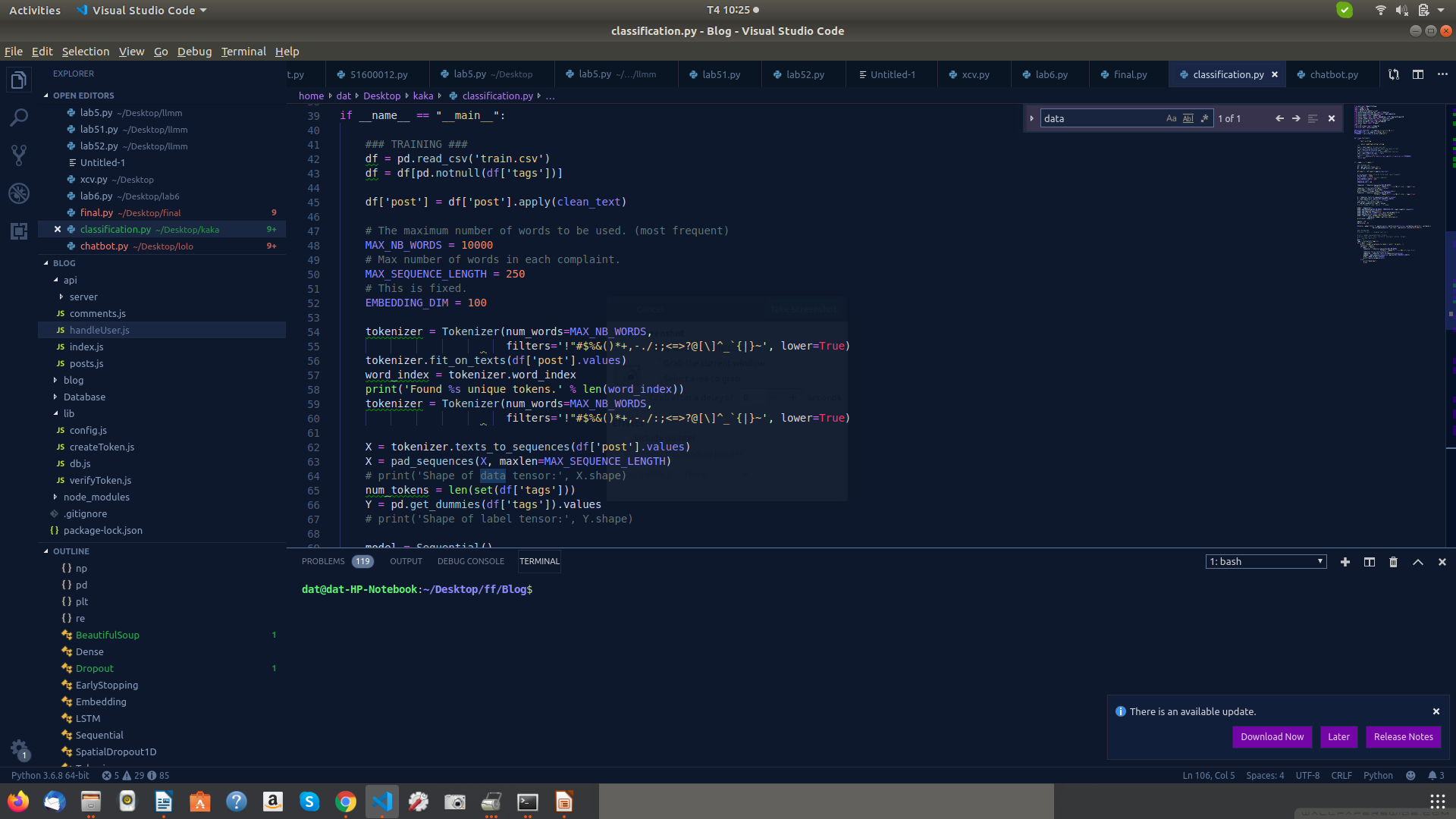
**Recurrent Neural Network**

**Code**

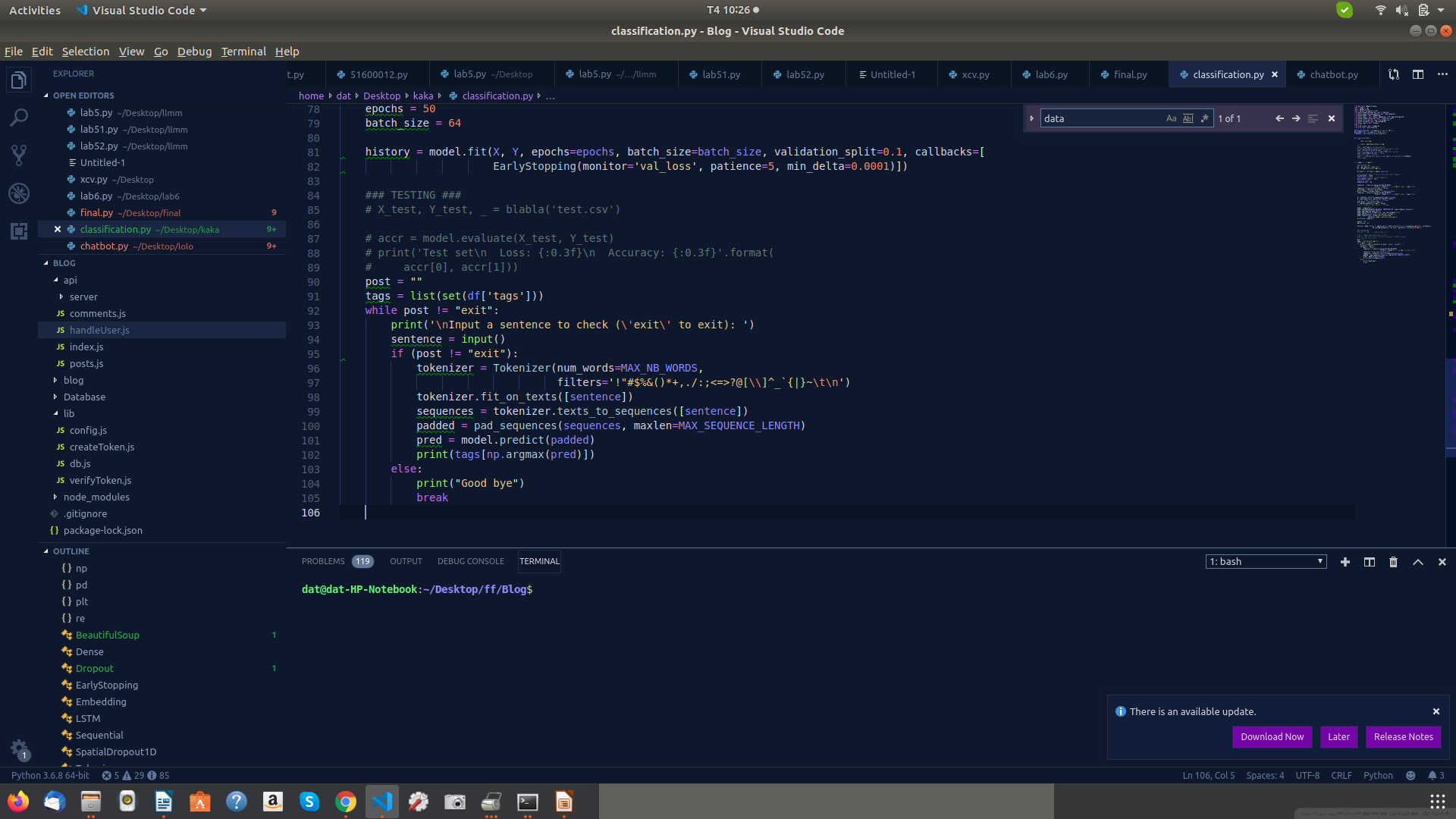
Classify

Executive data.



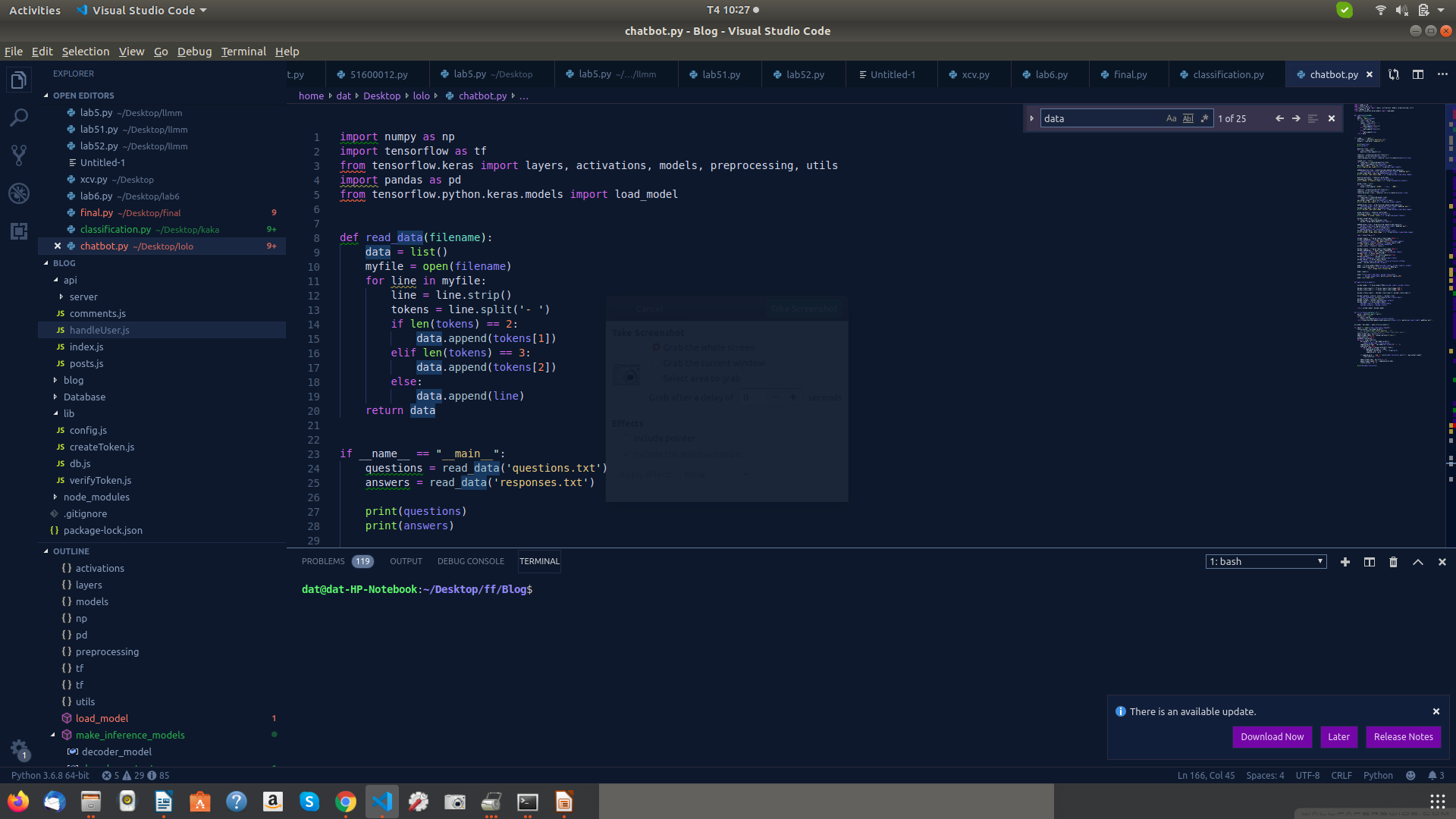
Trainning data

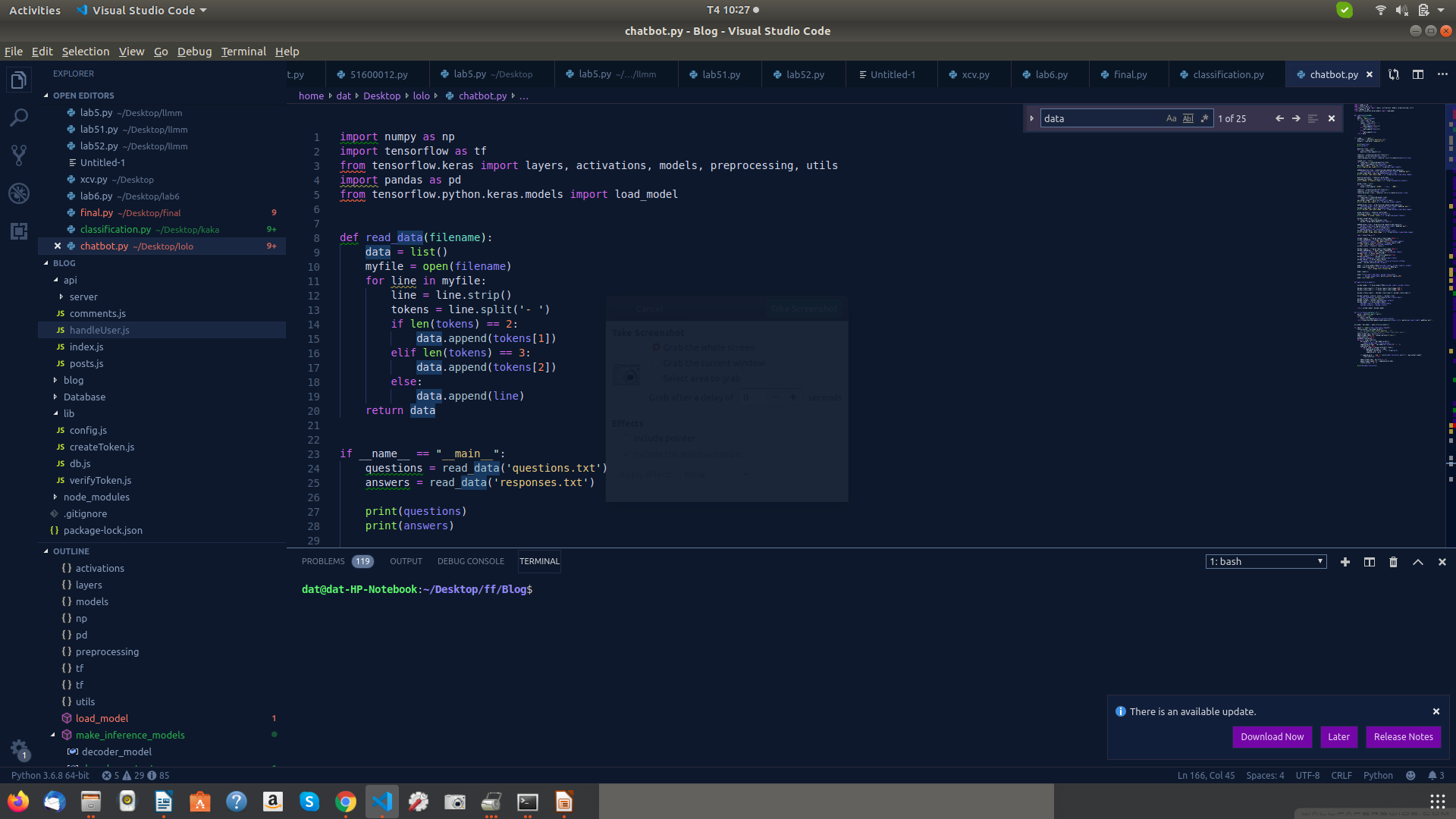
Tokenzier input



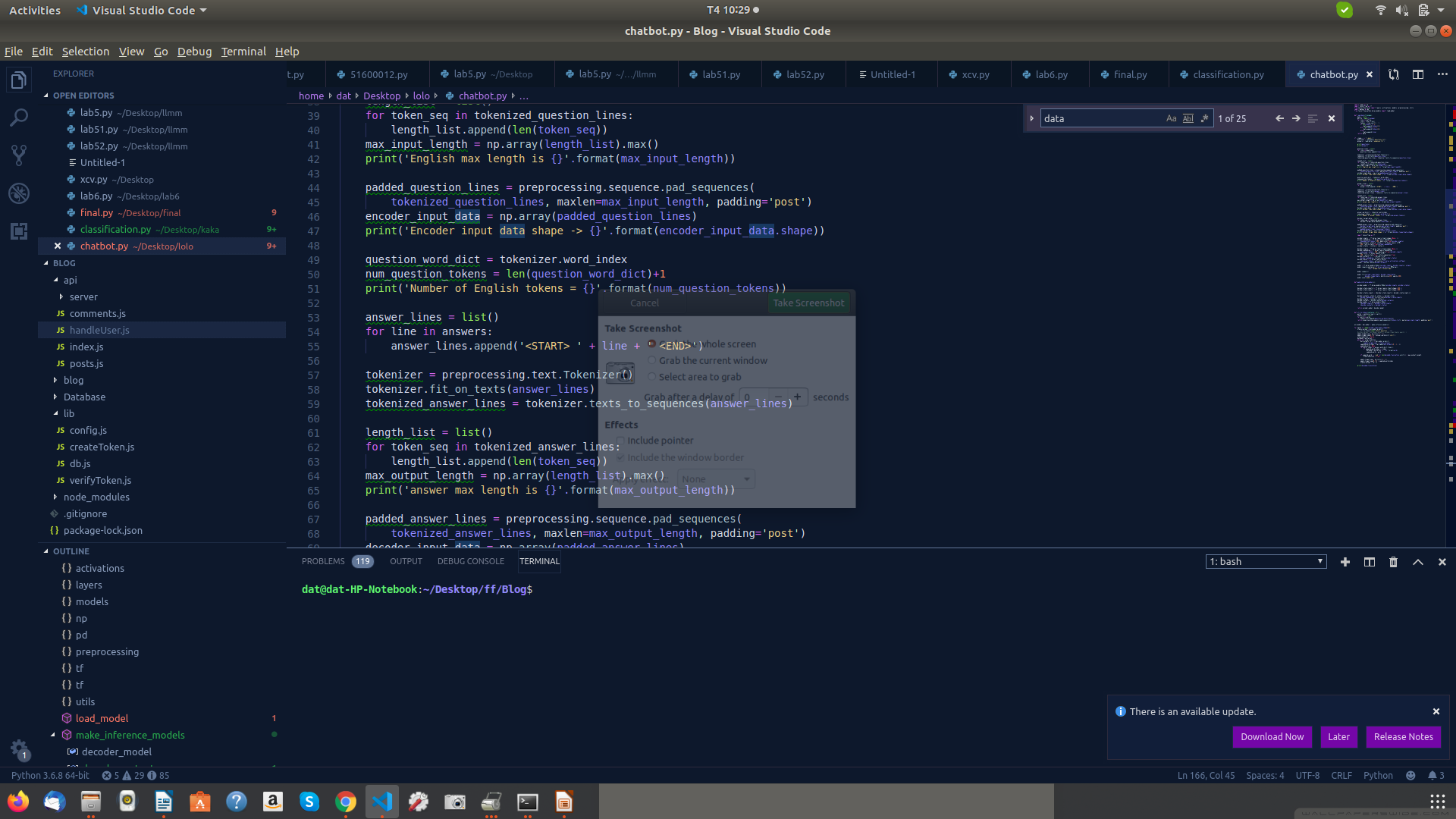
Chatbot

Executive data

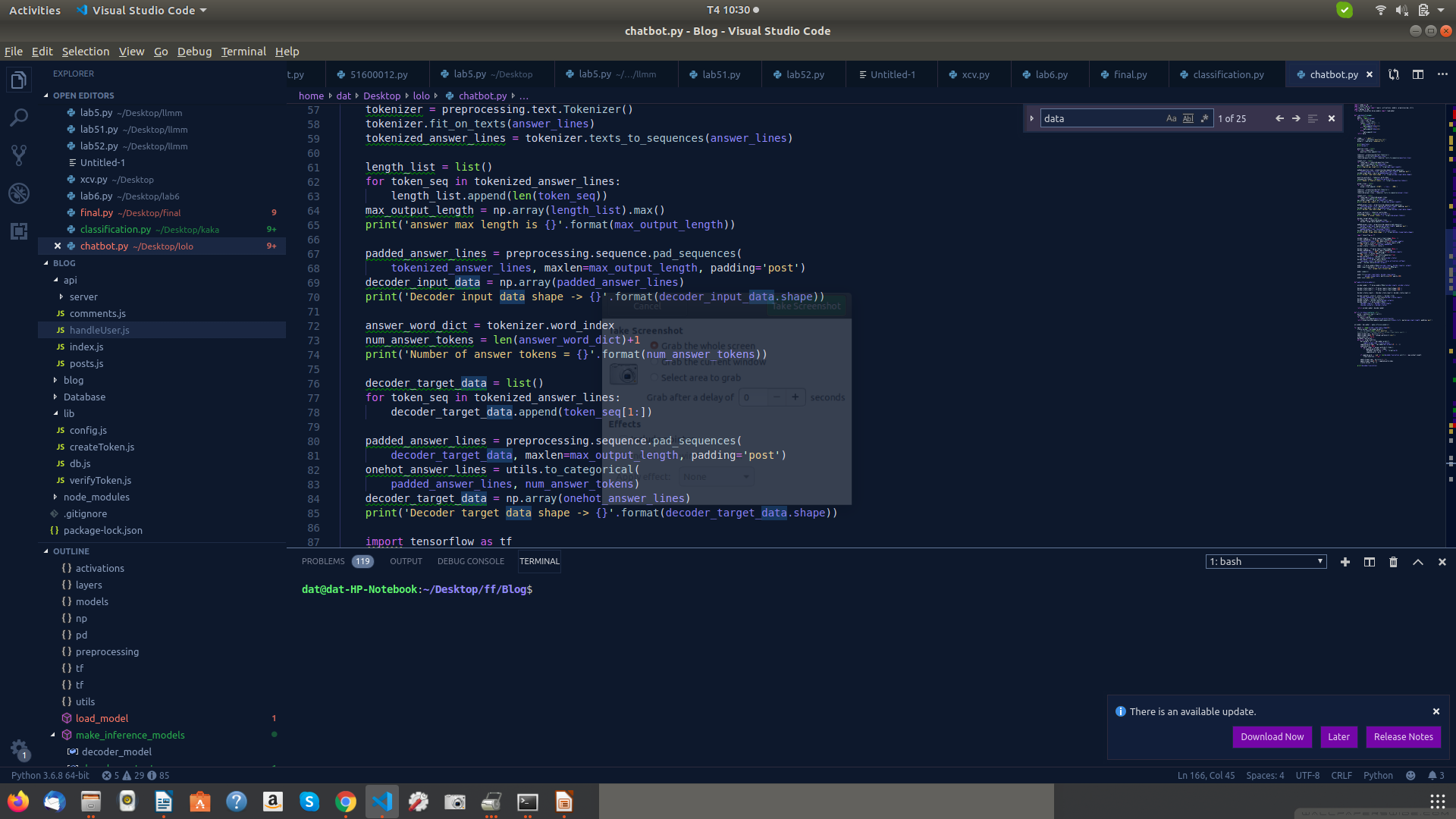


Encode phase

Tokenizer text



Decode phase



**Result**

