

Deep Learning bots

RNN: recurrent neural networks

- have been specifically designed to process inputs in a temporal order and update the future based on the past, as well as process sequences of arbitrary length
- RNNs became widely adopted within natural language processing because they integrate a loop into the connections between neurons, allowing information to persist across a chain of neurons.

LSTM: Long short-term memory networks

- Used in deep learning programs for natural language processing
- Chain of repeating neurons; each layer of neurons will perform a single operation on the input data
- There are two states that are produced for the first step in the sequence and then carried over as subsequent inputs are processed: cell state, and hidden state.
 - ⇒ cell state carries information through the network as we process a sequence of inputs.

- ⇒ At each time step updated input is appended to cell state by a gate which controls how much of the input should be included in the final product of the cell state.
- ⇒ This final product, which is fed as input to the next neural network layer at the next timestep, is called a hidden state.
- ⇒ The final output of a neural network is often the result contained in the final hidden state, or an average of the results across all hidden states in the network.
- ⇒ The persistence of the majority of a cell state across data transformations, combined with incremental additions controlled by the gates, allows for important information from the initial input data to be maintained in the neural network. Ultimately, this allows for information from far earlier in the input data to be used in decisions at any point in the model.

Seq2Seq: sequence-to-sequence model

- most common neural models used for text generation
- uses recurrent neural networks like LSTM to generate output.

- seq2seq can generate output token by token or character by character.
- In machine translation, seq2seq networks have an encoder accepting language as input and outputting state vectors and a decoder accepting the encoder's final state and outputting possible translations.
- This model has 2 parts
 - ⇒ Encoder: accepts language (or audio or video) input. The output matrix of the encoder is discarded, but its state is preserved as a vector.
 - ⇒ A *decoder* that takes the encoder's final state (or memory) as its initial state. We use a technique called "teacher forcing" to train the decoder to predict the following text (characters or words) in a target sequence given the previous text.

Packages and libraries →

Tensorflow: Used to conduct machine learning and deep neural networking. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-

of-the-art in ML and developers easily build and deploy ML-powered applications.

Keras: Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation.

- The core data structures of Keras are layers and models
- Layers: the basic building blocks of neural networks in Keras.

⇒ Input: used to instantiate a keras tensor (A tensor is a generalization of vectors and matrices to potentially higher dimensions)

⇒ Dense: densely connected neural network layer. It is most common and frequently used layer. Performs the following operation

Output = activation (dot (input, kernel) + bias)

- Input → Input data
- Kernel → weight data
- Dot → dot product of input and weight
- Bias → biased value used in machine learning to optimize the model

- ⇒ LSTM: Based on available runtime hardware and constraints, this layer will choose different implementations (cuDNN-based or pure-TensorFlow) to maximize the performance.
- Models: The Model is the core Keras data structure. There are two main types of models available in Keras: The Sequential model, and the Model class used with the functional API.
 - ⇒ Model: groups layers into an object with training and inference features.
 - ⇒ load_Model: Saved models can be instantiated via load_model. The models returned by load_model is a compiled model ready to be used

Numpy: Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

23 - 6 - 2020

Redant