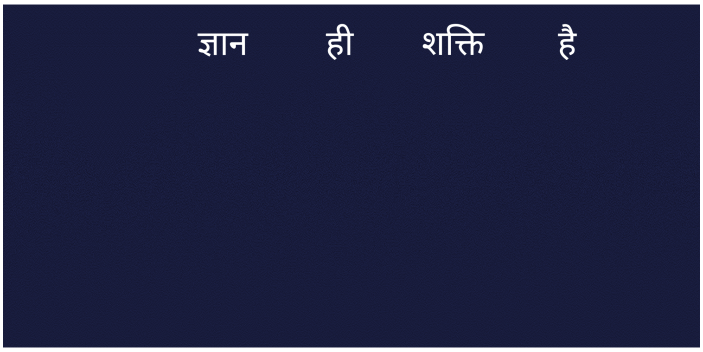
Deep Learning for Natural Language Processing

Generating text with seq2seq

The *seq2seq* (sequence to sequence) model is a type of encoder-decoder deep learning model commonly employed in natural language processing that 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.



One-hot vectors

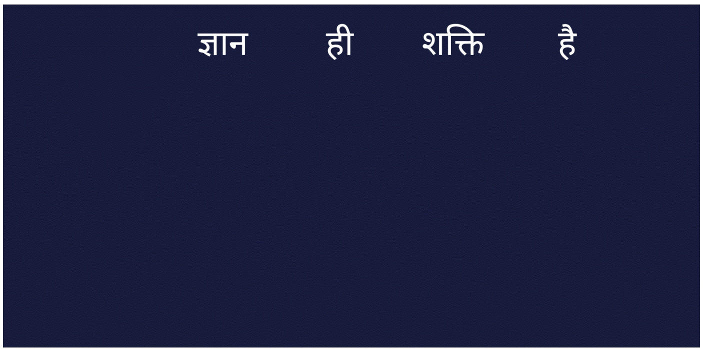
In natural language processing, *one-hot vectors* are a way to represent a given word in a set of words wherein a 1 indicates the current word and 0s indicate every other word.

```
# a one-hot vector of the word "squid"
# in the sentence "The squid jumped out
of the suitcase."

[0, 1, 0, 0, 0, 0, 0]
```

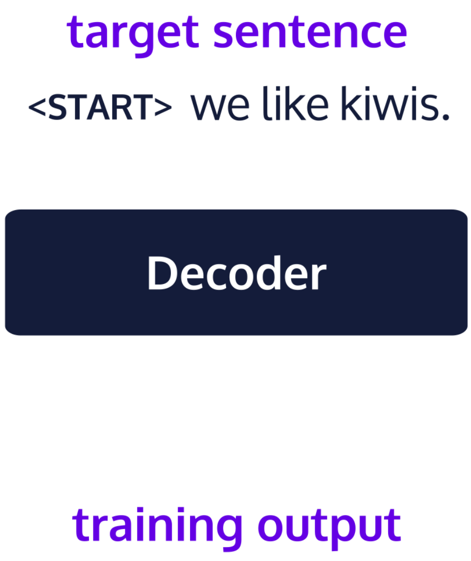
Seq2Seq Timesteps

For text generation, the neural *seq2seq* model needs to keep track of the current word being processed by its encoder or decoder. It does so with *timesteps*; each one indicates what token in a given document (sentence) the model is currently processing.



Teacher forcing for seq2seq

seq2seq machine translation often employs a technique known as *teacher forcing* during training in which an input token from the previous timestep helps train the model for the current timestep's target token.



Deep Learning with TensorFlow

Deep learning algorithms can be implemented in Python using the `TensorFlow` library, which is commonly used for machine learning applications such as neural networks. These can be created using `TensorFlow` with the `Keras API` .
To import the library:

```
from tensorflow import keras
```

The `layers` and `model` modules of `Keras` are used when implementing a deep learning model:

```
from keras.layers import Input,
LSTM, Dense
from keras.models import Model
```

Improving seq2seq

It is possible to improve seq2seq results by adjusting the model's quantity of training data, the dimensionality of hidden layers, the number of training epochs, and the training batch size.