2. Video captioning: Video can be seen as a sequence of images. Hence, in a video captioning problem, a sequence of images is passed as input to the LRCN model which in turn returns a sequence of outputs as a textual description for each video frame. Hence, video captioning can be seen as a many-to-many sequence problem. This approach is an example of an Encoder-Decoder LSTM where CNN is used as an image encoder that is initially trained for image classification. The final hidden layer, which is also called a bottleneck, is then passed as input to the RNN decode. It is typical to use an already pre-trained CNN on a large-scale image recognition task. A number of such models exist in the public domain. We will survey Encoder-Decoder LSTMs in more detail shortly. Video captioning is illustrated in Figure 36-17.

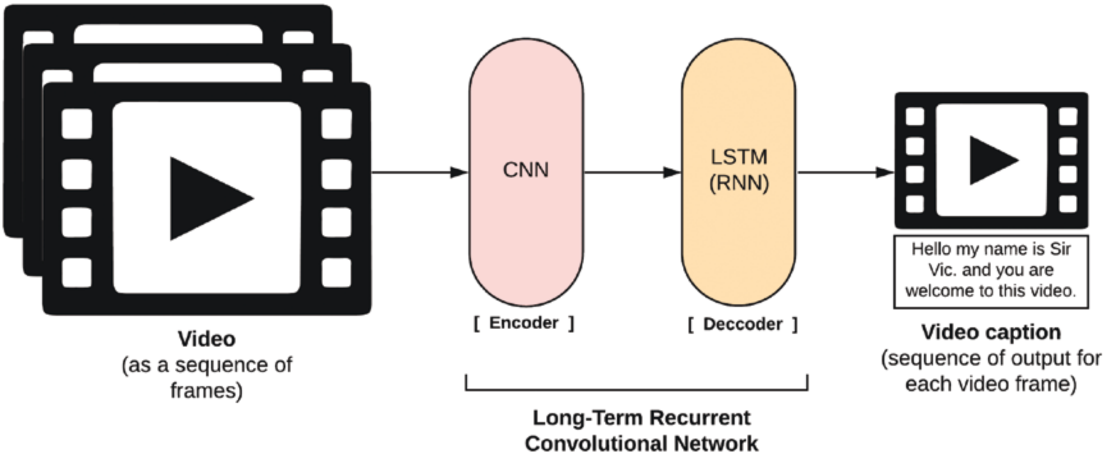
CNN
LSTM
(RNN)
Hello my name is Sir Vic. and you are welcome to this video.
Video
(as a sequence of frames)
[Encoder]
[Deccoder]
Video caption
(sequence of output for each video frame)
Long-Term Recurrent Convolutional Network

Figure 36-17. *Video captioning*

Encoder-Decoder LSTMs

Encoder-Decoder LSTM architecture handles a particular class of sequence problems that takes as input multiple time steps and also returns a multiple time step output. A major challenge of this sort of problems is that both the input and output sequences can have varied lengths.

The first part of the architecture, that is, the Encoder, is responsible for receiving and encoding the input sequence; the second part of the architecture, that is, the Decoder, takes in the output from the Encoder and then predicts the output sequence.

The sort of architecture is made for natural language processing problems where the output is a sequence of words. It is commonly used in machine translation, video captioning, and speech recognition. An illustration is already provided in Figure 36-10.

Bidirectional Recurrent Neural Networks

Bidirectional RNN is another particular type of recurrent neural network architecture that involves placing the recurrent layers beside each other where one layer works to learn the long-term dependencies from the past; this layer is called the forward LSTM. For the other layer, the input is reversed and fed into the network, so the network learns long-term dependencies from the future. This layer is called the backward LSTM. The bidirectional RNN is illustrated in Figure 36-18.

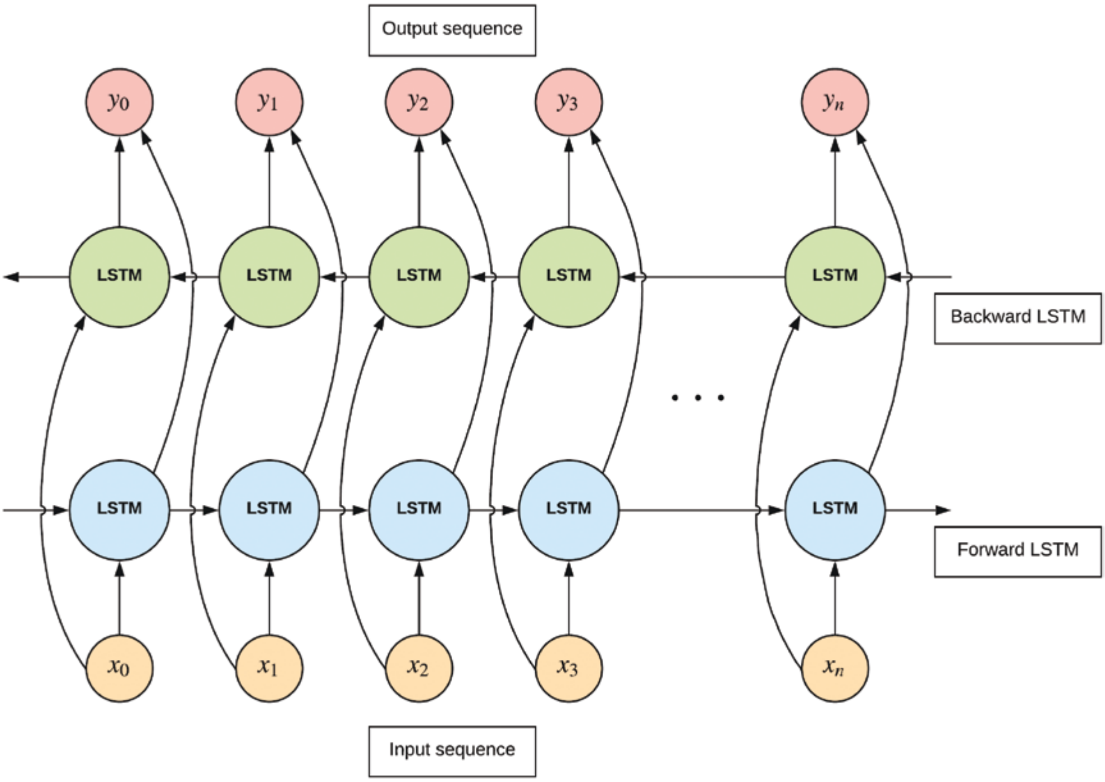

Figure 36-18. *Bidirectional LSTM*