**DL theory : Assingments-7**

1. Can you think of a few applications for a sequence-to-sequence RNN? What about a sequence-to-vector RNN, and a vector-to-sequence RNN?

* Sequence-to-sequence RNNs are often used in natural language processing tasks such as machine translation, text summarization, and image captioning.
* Sequence-to-vector RNNs are often used in tasks such as sentiment analysis, where the goal is to predict a single label for the entire sequence.
* Vector-to-sequence RNNs are often used in tasks such as speech synthesis, where the goal is to generate a sequence of audio samples based on a single input vector.

1. How many dimensions must the inputs of an RNN layer have? What does each dimension represent? What about its outputs?

* The inputs of an RNN layer must have three dimensions: (batch\_size, timesteps, input\_dim). The batch\_size dimension represents the number of instances in a batch, the timesteps dimension represents the number of time steps, and the input\_dim dimension represents the number of features at each time step. The outputs of an RNN layer also have three dimensions, with the same meanings.

1. If you want to build a deep sequence-to-sequence RNN, which RNN layers should have return\_sequences=True? What about a sequence-to-vector RNN?

* In a deep sequence-to-sequence RNN, all RNN layers except the last one should have return\_sequences=True. This is because the output of each RNN layer is passed as input to the next RNN layer, and we want to preserve the sequence dimension for all layers except the last one. In a sequence-to-vector RNN, the last RNN layer should have return\_sequences=False, so that the output is a single vector, instead of a sequence of vectors
* Suppose you have a daily univariate time series, and you want to forecast the next seven days. Which RNN architecture should you use?
* For a daily univariate time series, an architecture that can be used is a time-series forecasting model, such as an RNN with a single LSTM or GRU layer, with the input being the past n days' data, and the output being the forecast for the next 7 days.
* What are the main difficulties when training RNNs? How can you handle them?
* Some of the main difficulties when training RNNs include the vanishing/exploding gradient problem, and the difficulty of training long-term dependencies. These difficulties can be handled by using techniques such as gradient clipping, and using architectures such as LSTM or GRU that have mechanisms to handle long-term dependencies.
* Can you sketch the LSTM cell’s architecture?
* An LSTM cell has several components: an input gate, a forget gate, an output gate, a memory cell, and a hidden state. The input, forget, and output gates control the flow of information into and out of the memory cell, while the hidden state is used as the output of the LSTM cell.
* Why would you want to use 1D convolutional layers in an RNN?
* 1D convolutional layers can be used in an RNN to extract features from the input sequence. This can be useful for tasks such as speech or audio processing, where the goal is to extract features such as pitch, or spectral characteristics from the input sequence.
* Which neural network architecture could you use to classify videos?
* One neural network architecture that can be used to classify videos is a 3D CNN, which can extract spatiotemporal features from a video. Another architecture that can be used is a two-stream CNN, where one stream processes the RGB frames of the video, and the other stream processes the optical flow of the video.
* Train a classification model for the SketchRNN dataset, available in TensorFlow Datasets.