1. A recurrent neural network (RNN) receives sequences of 2-dimensional inputs and has three hidden neurons and one output neuron. The weight matrix $\boldsymbol{U}$ connecting the input to the hidden layer, the weight matrix $\boldsymbol{V}$ connecting the hidden output to the output layer, the hidden layer bias $\boldsymbol{b}$ and the output layer bias $\boldsymbol{c}$ are given by

$$\boldsymbol{U} = \begin{pmatrix} -1.0 & 0.5 \\ 0.5 & 0.1 \\ 0.2 & -2.0 \end{pmatrix}, \boldsymbol{V} = \begin{pmatrix} 2.0 \\ -1.5 \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}, \text{ and } \boldsymbol{c} = 0.5.$$

   Find the output sequence for an input sequence of $\left(\boldsymbol{x}(t)\right)_{t=1}^{4}$ where

$$\boldsymbol{x}(1) = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}, \boldsymbol{x}(2) = \begin{pmatrix} -1 \\ 1 \\ -2 \end{pmatrix}, \boldsymbol{x}(3) = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}, \text{ and } \boldsymbol{x}(4) = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix} \text{if}$$

   (a) The RNN is of hidden-recurrence (Elman) type with the recurrence weight matrix $\boldsymbol{W}$ connecting the previous hidden output to the current hidden layer input is given by

$$\boldsymbol{W} = \begin{pmatrix} 2.0 & 1.3 \\ 1.5 & 0.0 \end{pmatrix}.$$

   Assume that the hidden activations are initialized to zero.

   (b) The RNN is of top-down recurrence (Jordan) type with the weight matrix $\boldsymbol{W}$ connecting the previous output to the current state of hidden layer is given by

$$\boldsymbol{W} = (2.0 \quad 1.3).$$

   Assume that the output activations are initialized to zero.

2. An RNN receives 3-dimensional input sequence and produces 2-dimensional output sequence. It has 5 hidden neurons and receives sequences of 100 time steps.

   - Generate 8 training input sequences by drawing values uniformly between 0 and 1.0.
   - If the sequence $\left(\boldsymbol{y}(t)\right)_{t=1}^{100}$ where $\boldsymbol{y}(t) = \left(y_k(t)\right)_{k=1}^{2} \in \boldsymbol{R}^2$ denotes the output sequence for an input sequence $\left(\boldsymbol{x}(t)\right)_{t=1}^{100}$ where $\boldsymbol{x}(t) = \left(x_i(t)\right)_{i=1}^{3} \in \boldsymbol{R}^3$, generate the corresponding output sequences for the input training sequences as follows:

$$y_1(t) = 5x_1(t) - 0.2x_3(t-1) + 0.1\varepsilon$$
$$y_2(t) = 25x_2(t-1)x_3(t-3) + 0.1\varepsilon$$

     where ε is standard normally distributed random variable.

   Train an RNN to learn the above sequences by using gradient descent learning. Use a learning factor $\alpha = 0.01$ and Adam optimizer.