Q. 1 Consider an input image of size 13×13 and 64 filters of size 3×3. Discuss whether it is possible to perform convolutions with strides 2, 3, 4, and 5. Justify your answer in each case.

The answer depends on the padding of the input image. Padding refers to the technique of adding extra pixels around the edges of the input image to allow for the filter to perform convolutions at the edges of the image.

If the input image has zero-padding around the edges (i.e., the size of the output feature map is the same as the size of the input image), then it is possible to perform convolutions with strides of 2, 3, 4, and 5.

Stride 2: If we apply a stride of 2 to the convolution operation, the output feature map would be of size **6x6**.

Stride 3: If we apply a stride of 3 to the convolution operation, the output feature map would be of size **4x4**.

Stride 4: If we apply a stride of 4 to the convolution operation, the output feature map would be of size **3x3**.

Stride 5: If we apply a stride of 5 to the convolution operation, the output feature map would be of size **2x2**.

However, if the input image does not have any padding and is smaller than the filter size, then convolutions with strides 2, 3, 4, and 5 will not be possible as the filter will not be able to perform convolutions at the edges of the input image

Q. 2 Differentiate between CNN and RNN with suitable examples.

S.no	CNN	RNN
1	**CNN** stands for **Convolutional Neural Network**.	**RNN** stands for **Recurrent Neural Network**.
2	CNN is ideal for images and video processing.	RNN is ideal for text and speech Analysis.
3	It is suitable for spatial data like images.	RNN is used for temporal data, also called sequential data.
5	The network takes fixed-size inputs and generates fixed size outputs.	RNN can handle arbitrary input/ output lengths.
6	CNN is a type of feed-forward artificial neural network with variations of multilayer perceptron's designed to use minimal amounts of preprocessing.	RNN, unlike feed-forward neural networks- can use their internal memory to process arbitrary sequences of inputs.
7	CNN's use of connectivity patterns between the neurons. CNN is affected by the organization of the visual **cortex**, whose individual neurons are arranged in such a way that they can respond to overlapping regions in the visual field.	Recurrent neural networks use time-series information- what a user spoke last would impact what he will speak next.

Q.3 What are the major issues in RNN? Discuss the methods to address these issues.

Recurrent Neural Networks (RNNs) are a type of neural network that can process sequential data such as time series data or natural language. Despite their success in various applications, RNNs suffer from some issues. In this answer, we will discuss some major issues in RNN and methods to address these issues.

Vanishing and Exploding Gradients:

The issue of vanishing and exploding gradients occurs when the gradients in the backward pass either become too small or too large. This leads to the weights of the network being updated too little or too much, which can cause the network to learn poorly or not at all. The problem becomes more severe with deeper networks and longer sequences.

To address this issue, various techniques have been proposed, including:

Gradient Clipping: A technique that involves clipping the gradients to a maximum value to prevent them from becoming too large.

Weight Initialization: Initializing the weights in a way that keeps the gradients from vanishing or exploding.

Using Different Activation Functions: Some activation functions like ReLU and its variants can help alleviate the vanishing gradient problem.

Using Different RNN Architectures: Some RNN architectures like LSTM or GRU are designed to mitigate the vanishing gradient problem.

Memory Limitations:

The standard RNN architecture suffers from memory limitations and can only keep track of short-term dependencies. This makes it difficult for the network to learn long-term dependencies.

To address this issue, various architectures have been proposed, including:

Long Short-Term Memory (LSTM): A type of RNN architecture that can store information for long periods of time and selectively forget information that is no longer needed.

Gated Recurrent Unit (GRU): A simplified version of LSTM that also has a gating mechanism to control the flow of information.

Training Time:

Training RNNs can be computationally expensive, especially for large datasets and deep networks. This can make it difficult to train the network efficiently.

To address this issue, various techniques have been proposed, including:

Mini-Batch Training: Using mini-batches of data to update the weights of the network, this can speed up the training process.

Dropout: Regularization techniques that can help prevent over fitting and speed up training.

Early Stopping: Stopping the training process when the validation loss stops improving, this can prevent over fitting and save time.

In conclusion, while RNNs have been successful in various applications, they suffer from some issues. By using techniques such as gradient clipping, weight initialization, different activation functions, LSTM, GRU, mini-batch training, dropout, and early stopping, these issues can be addressed and the performance of RNNs can be improved.

Q. 3 What is the difference between the workflow of LSTM and GRU? Give the examples where LSTM should be used over GRU.

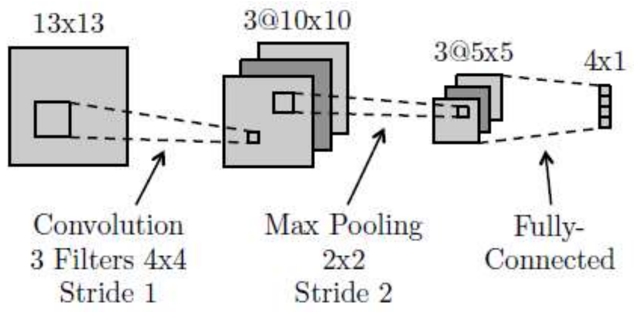
LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) are both types of recurrent neural networks (RNNs) that are commonly used for sequential data modelling. The key difference between LSTM and GRU lies in the way they handle memory.

LSTM has a more complex memory structure, with a memory cell that is controlled by three gates: the input gate, forget gate, and output gate. These gates allow the LSTM to selectively

update, forget, and output information from the memory cell, which makes it better suited for tasks that require the model to retain information over longer periods of time.
GRU, on the other hand, has a simpler memory structure, with just two gates: the reset gate and the update gate. The reset gate controls how much of the previous hidden state should be forgotten, while the update gate controls how much of the new information should be added to the hidden state. This simplicity makes GRU models faster to train and more computationally efficient, which can be advantageous in certain scenarios.
Examples where LSTM should be used over GRU:
Language modelling: Tasks that involve modelling long sequences of text, such as language modelling, may require a more complex memory structure like that of an LSTM to handle long-term dependencies and retain important information over longer periods of time.
Speech recognition: Speech recognition is another task that requires the model to retain information over a longer period of time. LSTM models have been shown to be effective in this domain.
Video analysis: Tasks such as video analysis, where the model needs to retain information over a longer time period, would require LSTM over GRU as it can handle long-term dependencies.

In summary, LSTM is better suited for tasks that require the model to retain information over longer periods of time and handle long-term dependencies, whereas GRU is a more computationally efficient option that may be better suited for tasks that require faster training or have limited computational resources.

Q. 4



Answer following questions about the above network:

i. How many weights in the Convolution layer do we need to learn?
48 weights. Three filters with 4x4=16 weights each

ii. How many ReLU operations are performed in the forward pass?
75 ReLu operations. ReLu is performed after the pooling step. ReLu is performed on each pixel of the three 5x5 feature images.

iii. How many weights do we need to learn for the entire network?
348 weights. 48 for the convolutional layer. Fully-connected has 3x5x5=75 pixels each connected to four outputs, which is 300 weights. Pooling layer does not have any weights.