Question 3: Compare the performance of the LSTM RNN network vs CNN.

**Answer:**

* RNN can model sequences of data so that each sample can be

assumed to be dependent on previous ones.

* CNNs are commonly used in solving problems related to spatial

data, such as images. RNNs are better suited to analyzing temporal, sequential data,

such as text or videos.

* CNNs are "feed forward neural networks" that use filters and

pooling layers, whereas RNNs feed results back into the network.

* In CNNs, the size of the input and the resulting output are

fixed. That is, a CNN receives images of fixed size and outputs

them to the appropriate level, long with the confidence level of

its prediction. In RNNs, the size of the input and the resulting output may vary.

* CNN is faster than RNN, because CNN is nothing to have store in a memory

as RNN.

To achieve the same test performance CNN is taking less time than LSTM RNN. The time require to complete each iteration is more in RNN. CNN is faster than RNN.

As CNN are good for fixed length input data like images, facial recognition, medical analysis, and classification.

LSTM RNN is good for text translation, natural language processing,

sentiment analysis and speech analysis. Training the RNN is a very difficult task.

It cannot process very long sequences if using tanh or relu as an activation function.

In LSTM, Pre-processing can increase the score by providing high-quality input data. Cross Validation detects and prevents overfitting and the decrease of score caused by overfitting.

Question 4: summary on how RNN and CNNs can be combined for the classification task

**Answer:**

Deep convolutional neural networks (CNNs) are proven to be good with single -label image classification. The combination of CNN-RNN framework learns a combined image-label embedding to characterize the semantic label dependency and the image-label relevance. This combined model can be used to train from the beginning to the end to integrate the joint information.

Real world images contain multiple labels, which could correspond to different objects, scenes, actions, and attributes in an image. And to understand the image, modeling semantics and the dependencies are important.

CNNs to multi-label classification is to transform it into multiple single-label classification problems, which can be trained with the ranking loss or the cross-entropy loss. when we work on the labels separately, these methods will fail to create dependency between multiple labels. multi-label classification problems exhibit strong label co-occurrence dependencies.

When dealing with a large set of labels, the parameters of these pairwise probabilities can be prohibitively large while lots of the parameters are redundant if the labels have highly overlapping meanings.

RNN improves the Classification accuracy to model the label dependencies to capture higher-order label relationships while keeping the computational complexity tractable.

For CNN, there is a problem of overfitting, as previous methods assume all classifiers share the same image features. But the flaw in the method is, while doing these small objects get ignored or difficult to recognize individually. To overcome this flaw RNN can add the extra points for adapting the feature of images based on the previous predictions. The motive is to implicitly adapt the attentional area in images so the CNNs can focus its attention on different regions of the images when predicting different labels. The many image labels improve the generalization ability because the labels with duplicate semantics can get more training data. Also utilizing the semantic redundancies reduce the computational cost.

CNN-RNN framework for multi-label image classification effectively learns both the semantic redundancy and the co-occurrence dependency in an end-to-end way. The multi-label RNN model learns a joint low-dimensional image-label embedding to model the semantic relevance between images and labels. The image embedding vectors are generated by a deep CNN while each label has its own label embedding vector. The high-order label co-occurrence dependency in this low-dimensional space is modeled with the long, short term memory recurrent neurons (LSTM), which maintains the information of label context in their internal memory states.

The CNN-RNN model combines the advantages of the joint image/label embedding and label co-occurrence models. Also gives better performance compared to the current state-of-the-art multi-label classification methods. Like humans’ multi-label classification process, RNN framework can focus on the corresponding image regions when predicting different labels.

The CNN part extracts semantic representations from images; the RNN part models image/label relationship and label dependency. The RNN model is employed as a compact yet powerful representation of the label co-occurrence dependency in this space. It takes the embedding of the predicted label at each time step and maintains a hidden state to model the label co-occurrence information. The a priori probability of a label given the previously predicted labels can be computed according to their dot products with the sum of the image and recurrent embeddings.

CNN-RNN model can be learned using cross-entropy loss on the SoftMax normalization. The paper results include easier objects that should be predicted first to help predict more difficult objects. Also learning label orders by iteratively finding the easiest prediction ordering and order ensembles as proposed in or simply using fixed random order, but they do not have notable effects on the performance.

In conclusion, CNN-RNN model also effectively learns a joint label/image embedding. CNN-RNN framework combines the advantages of the joint image/label embedding and label co-occurrence models by employing CNN and RNN to model the label co-occurrence dependency in a joint image/label embedding space.