## Sprint Assessment

Question 1

Why are recurrent neural networks (RNN) especially suited for modeling sequential data?

* They operate on sequences of data and consider all of the inputs given to the model in the past.
* Sequential data requires special modeling and RNNs are the only suitable machine learning model type.
* They accept as input only time-series data but are very good at predicting the next data point in the sequence.
* A RNN is the always the computationally fastest way to model sequential data.

Question 2

What does the term "recurrent" mean as used in "Recurrent Neural Network" (RNN)?

* A RNN uses convolution recursively to update the weights of the hidden layers.
* A RNN can "remember" things learned from prior input; the generated output is copied and sent back into the input, which is a recursive or recurrent process.
* The training data is input into a RNN using a recursive process, where the data is updated depending on the output from the hidden layer.
* All neural network use a recurrent process when they update the weights in the hidden layer so it's just a preference as to what the process is called.

Question 3

The term "convolution" is best defined as:

* A mathematical operation where one function is modified by a second function to produce a third output function which describes that modification.
* A special mathematical function that can also be used as a loss function.
* A mathematical operation that is used only with convolutional neural networks and doesn't have any other applications.
* A mathematical operation that sounds complicated by is really just the multiplication of two functions resulting in a new output function.

Question 4

What is the class of task your Autoencoder is best suited for?

* Representation of data
* Clustering
* Classification of data
* Semi-supervised learning

## RNN

Question 5

What is recursive about a recurrent neural network (RNN)?

* There is a recurrent loop where the output from a hidden later is "recycled" and added back in as input into that layer.
* The process of feeding the final output of the RNN back into the input layer.
* The weights applied to the input are recycled at the hidden layers and fed back in to prevent the vanishing gradient problem.
* The recurrent part of the neural network is how often each neuron is "recycled" or used again in the next hidden layer.

Question 6

Why are RNNs and their subsets (LSTM networks) suitable for modeling time or sequence-dependent data?

* The input layer in a RNN accepts sequences and can then pass long sequences into the recurrent layer.
* A RNN trains faster because of the recurrent hidden layer so long input sequences are easier to remember.
* RNNs use a large number of nodes so they can model long sequences of input.
* An RNN maintains internal memory of earlier input, which is especially important when modeling sequences.

Question 7

What problem does an LSTM solve, when compared to a RNN?

* vanishing gradient
* decreasing cross entropy
* increasing bias
* vanishing dropout

Question 8

What are the sequences we can use when predicting text using an LSTM RNN?

* Sequences of tokens.
* Sequences of individual characters.
* Sequences of individual characters and tokens (words).
* Vectorized sentences.

## CNN

Question 9

Which of the following choices is the best description of convolution as it relates to an image?

* The process of compressing an image so it is easier to use as input to a convolutional neural network (CNN).
* The process of convolution is taking the output from a hidden layer and feeding it back in as input.
* Convolution is the process of sliding a filter across an image in order to produce a new matrix.
* Convolution is a different way to describe the process of edge detection in an image.

Question 10

What is the purpose of the convolutional layer in a CNN?

* It reduces the dimensions of the convolved images to reduce the training time.
* It creates feature maps that represent the features that the layer has learned, which are used by the fully connected portion of the model to make better predictions.
* It uses filters to prevent the vanishing or exploding gradient problem associated with CNNs.
* It accepts the filtered or convolved initial input and passes it to an additional filtering layer.

Question 11

How does the pooling layer work in a CNN?

* It slides a window over the image and only keeps the max, min, or average value in that window in order to reduce the size of the image.
* It combines separate convolutional filters into one "pooled" filter to speed up processing.
* It combines pixels in the input image into one pixel, with either the average or maximum value.
* It combines or pools the results from numerous filters in the convolutional layers into the final model parameters.

Question 12

Consider the following convolutional layer (added with the Keras sequential API):

*model.add(Conv2D(32, (3, 3), activation='relu',*

*kernel\_initializer='he\_uniform',*

*input\_shape=(28, 28, 1)))*

What is the shape of the kernel filter and how many filters are in this layer?

* kernel\_size = 28x28, filters = 32
* kernel\_size = 9, filters = 28x28x1
* kernel\_size = 3x3, filters = 32
* kernel\_size = 32, filters = 9 (or 3x3)

Question 13

What type of process are we taking advantage of when we use a pre-trained neural network to classify images?

* transfer learning
* categorical cross-entropy
* activation
* convolution

## AutoEncoder

Question 14

What type of neural network is an autoencoder?

* perceptron
* recurrent
* LSTM
* encoder-to-decoder

Question 15

What is the process by which noise is reduced when using an autoencoder?

* An autoencoder uses a recurrent layer to filter the data, reducing the noise. It works in a similar way to a recurrent neural network.
* An autoencoder learns a compressed representation of the input by implementing dimensional reduction, which reduces noise.
* In order to use an autoencoder, the dimensions of the input data need to be reduced before feeding it to the input layer of the autoencoder.
* The output from an autoencoder is filtered to reduce the noise.

Question 16

What type of process does an autoencoder use to learn the encodings?

* logistic regression
* supervised learning
* clustering
* unsupervised learning

Question 17

If the input dimension is 784, what is the output dimension from an autoencoder?

* 784
* 32
* 28
* 2

Question 18

If an autoencoder is trained on images of digits, will it be able to encode others types of images, such as landscape or animals?

* No but this is because autoencoders don't work well for modeling images.
* Yes. As long as the autoencoder was trained on images, it will work well for all image types.
* No. It learns how to encode a specific type of image and isn't useful for images that are of a very different type.
* Yes but only if we train the autoencoder specifically on images of digits, as this specific type generalizes well to other images.

Question 19

What is the process of information retrieval when using an autoencoder?

* Train an encoder, return all the images that were encoded, find an image that best matches the one you want to retrieve
* Set-up the autoencoder architecture, select images that are similar to what you want to retrieve, train on those images, return the retrieved image
* You can't retrieve anything from an autoencoder because it doesn't produce output that is much different from the input.
* Train an autoencoder, select an image to retrieve (query); use k nearest-neighbors to retrieve an image similar to the query

## AI Future

No questions