Q1.

What is "Deep Learning?" How does it different from traditional machine learning techniques such as SVMs, Naive Bayes, Decision Trees, etc? Why has it become so popular in the last decade?

Ans.

Deep Learning is based on data representation as opposed to learning based on tasks like in other machine learning techniques such as SVM, Naïve Bayes etc. This basically means that each feature is given specific weights in numeric values and channeled through several hidden layers and adjusted according to the target values. The main reason it's become popular is because it's easy and it works! Specially with image data the classification accuracy has surpassed previous boundaries now that we have high processing computers.

Q2.

What are activation functions? What is their purpose? Must they be non-linear? Must they be continuously differentiable? Must they be monotonic? How does the derivative of the activation function effect the gradient?

Ans.

Activation functions are basically a switch which turns on based on the input. Each node is given an activation function and based on the weights and biases calculates whether that node should be included in the future analysis or not. They **must** non-linear so that the combination of their weights has certain effect on the nodes otherwise the transformation is just scaled and doesn't function like a switch. They must not be monotonic otherwise error associated comes out to be convex. The activation function should be continuously derivable to enable gradient based optimization methods.

Q3.

What is backpropagation? Name three backpropagation algorithms and explain how they work. List the pros and cons of your three backpropagation algorithms.

Ans.

Backpropagation is the weight of current node decided by the result of the previous node. Suppose in a hidden layer with the same feature importance the word was 's' and we know via dictionary that 'z' never comes after s in any word so the weight for that node will be very low.

Pros:

1. Very Flexible, can be used for pattern recognition as well as decision making.
2. Parallel processes make it faster to run on modern computers.
3. It gives very accurate predictions and results.

Cons:

1. It can take long time to process to train, once it's trained it's very quick
2. Size of the training data must be large for accurate prediction
3. Tuning the activation function by trail-error can be time taking.

Q4.

Name two forms of regularization used in neural networks. Explain how they work. Why does one use regularization with neural networks?

Ans.

L1 and L2 (also dropout, data augmentation) are the regularizations used in neural networks. They're basically used to prevent overfitting of the data by penalizing the coefficients in machine learning, in deep learning **it penalizes the weight matrices of the nodes.** We just update the general cost function by adding another term known as the regularization term in neural networks.

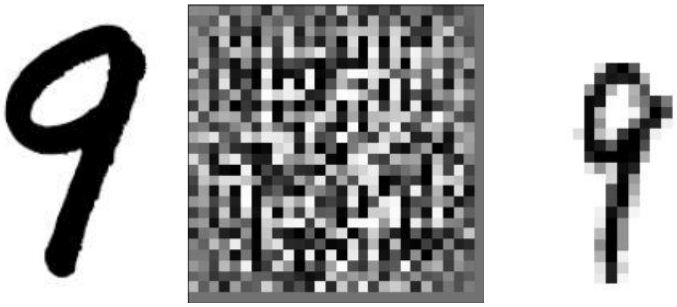
Q5.

What is a convolution? Give an example of a convolution on image data with and without padding.

Ans.

convolution parameters consist of a set of learnable filters. Every filter is small with width and height but extends through the full depth of the input volume. Suppose an image with dimensions 5X3X3 with passing of the convolutional layer we create a 2-dimensional activation map that gives the responses of that filter at every position.

Here are the examples



Q6.

What is max-pooling? Why is it used? How does it compare to mean-pooling and min-pooling? Which would you use between max-pooling, mean-pooling and min-pooling?

Ans.

We want to converge the weight values suppose from 2X2 matrices to a single value, we want that value to represent the whole matrix. This is done to forma non-linear down sampling. Max pooling is taking the largest value of that matrix, compared to min value if only one value is 0 then the whole matrix becomes 0 which doesn't represent the matrix. We can use mean pooling as well and it'd work almost the same as max pooling.

Q7.

What is the feature learning pipeline in a CNN? How does it relate to the classification in a CNN? Can the feature learning pipeline be used with techniques like SVMs, Naive Bayes, Decision Trees, or GBMs?

Ans.

Feature learning pipeline is a set of functions that each input needs to be passed through for a feature. Like a pipeline could be input -> Convolution -> max_pooling-> activation function -> flatten -> softmax. In a CNN after convolution the pipeline usually has bunch of pooling and activation function to finally classify the input. Feature learning pipeline cannot be used by other machine learning techniques because they work on the principles of task of the input and not weights.

Q8.

What are loss functions? When would one choose cross-entropy vs mean-square error?

Ans.

Loss function is mapping of the features into a real number essentially to a number giving it a cost. So, if you have some inputs in an event that event costs a number and depending on that number being high and low that event is classified. Cross-entropy is preferred for **classification**, while mean squared error is one of the best choices for **regression**.

Q9.

What is an RNN? What kind of data is an RNN used for? How does a Vanilla RNN differ from an LSTM?

Ans.

A *recurrent neural network* (*RNN*) is a class of artificial neural network where connections between nodes form a graph along with a sequence. RNN is usually used for dictionary or word-sentence creating sort of data. A vanilla RNN is neural net with only one hidden layer a single layered perceptron, LSTM consists of a LSTM unit with features like cell, input gate, output gate and forget gate.

Q10.

What is a Markov model? What kind of data is a Markov model used for?

Ans.

Markov Model is a system in which data is assumed to follow Markov process in which the state is not directly visible, but the output dependent on the state, is visible. Dataset usually used are hard writing, gesture recognition, speech and part-of-speech tagging.

Q11.

What is network initialization? How can it effect a neural network? Name three common approaches for network initialization and why one would choose one over another.

Ans.

Network initialization refers to the shape and structure of the hidden layer matrices. Depending on the size of the hidden layers and epochs employed the training of the model could become more accurate and time consuming, if regularization is not done we could overfit the model.

1. Initializing the network with random weights
2. Setting all the weights in the network to 0.
3. Use ReLU as the activation function

Q12.

What is transfer learning? When would one use it? Give an example of transfer learning using neural networks.

Ans.

Transfer learning is using the knowledge gained by solving on problem and applying it on some different but unrelated problem. This could be used when we think that something similar is already done in another dataset which was perceived to solve this one. The results of the number dataset could be used in identification of objects in an image. Essentially same principal is applied in both datasets, but problems are vastly different.

Q13.

What is an autoencoder? What are they used for? Explain how an autoencoder works.

Ans.

It is a sort of unsupervised clustering of data to reduce the dimensions of the data. It has two parts encoder and decoder. The input is encoded and reduced to perform code operations and output is evaluated using the decoder.

Q14.

What is a variational autoencoder? What are they used for? How do they differ from autoencoders?

Ans.

Variational autoencoders are similar to auto encoder but they also factor in the distribution of latent variables which results in additional loss component. They are used when dataset follows a probabilistic model for which a graph expresses the conditional dependence structure between random variables.

Q15.

What are deep generative models? What are they used for?

Ans.

Deep generative models are essential in learning any kind of data distribution using unsupervised learning. The two main approaches for it are Variational Autoencoders (VAE) and *Generative* Adversarial Networks (GAN). Generative models aim at learning the true data distribution of the training set to generate new data points with some variations.

Q16.

What are multilayer perceptron? What are they used for?

Ans.

It is a feed forward artificial neural network and should contain at least 3 layers of nodes. It is used when the data is not linearly separable. They're useful in research for their ability to solve problems which contains random variables, which often allows approximate solutions for extremely complex problems.

Q17.

In deep generative models what is meant by density estimation? What is meant by latent variable models?

Ans.

Deep generative models are form of unsupervised learners, so each point of data grouped should have an importance in terms of its density. The higher the density more important the feature is. Latent variable models use auto encoders to convert input to smaller dimensional representation which can store latent information about the input data distribution.

Q18.

What are soft-max functions? What are they used for?

Ans.

Towards the end of output the probabilities for each classifier is not based on anything so we don't know which class is the optimal one, soft-max functions convert all the values to same base so we can relate them.

Q19.

What is a session in TensorFlow? Why aren't sessions required for scikit-learn?

Ans.

Session in a tensor flow records the state of each tensor unit for that particular time frame because they keep changing. Scikit-learn never does that so it doesn't require a session.

Q20.

TensorFlow uses a dataflow graph. What is it? Why is it used?

Ans.

Tensor board is a graphical representation of each hidden layer and the pipeline of the nodes through which input is processed. In a particular neural network we may not be aware of what a particular hidden layer is doing, session graphs help us understand those layers better. For example, in image classification during convolution phase the image may be distorted so bad that any further pipeline function couldn't classify the image comprehensibly.