Neural Networks

Final All Theory

Q1: Discuss Radial Basis Function Network, Self-Organizing Maps, and Generative Adversarial Network (GAN) with respect to architecture, activation function and applications.

Solution:

Radial Basis Function Network(RBFN)

Radial basis function (RBF) networks are a commonly used type of artificial neural network for function approximation problems.

Architecture: Radial basis function (RBF) networks typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer.

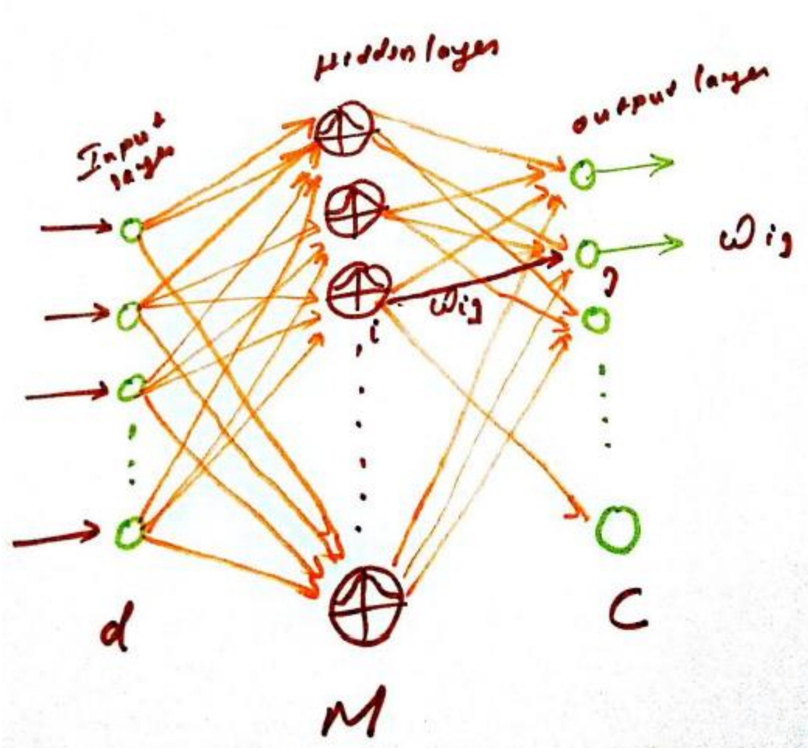
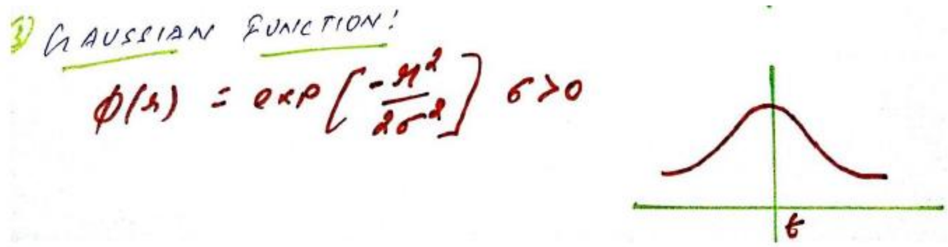


Figure1: Architecture of RBFN

Activation function:



Applications:

1. RBF Networks for Classification.
2. The XOR Problem in RBF Form.
3. Real World Application – EEG Analysis.

Self-Organizing Maps(SOM)

The Self-Organizing Map is one of the most popular neural models. It belongs to the category of the competitive learning network.

Architecture:

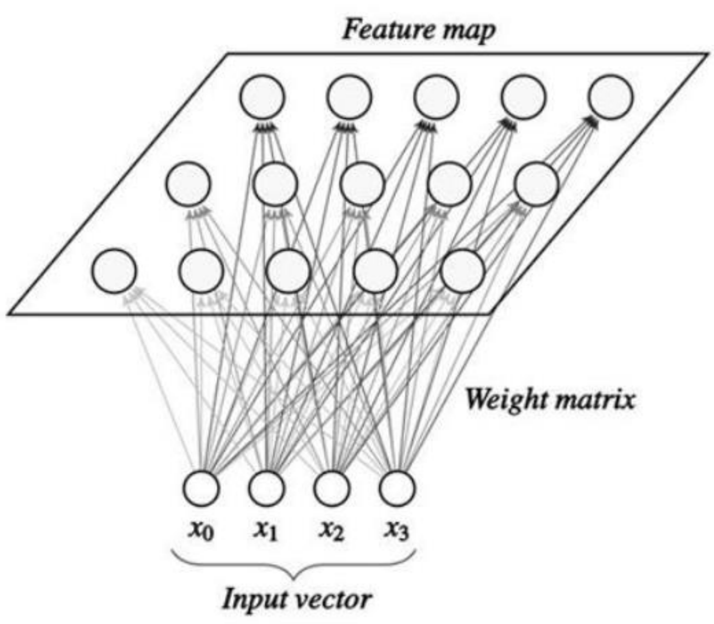


Figure 2: Architecture of SOM

Activation function: Unlike other ANN types, SOM doesn't have activation function in neurons. SOM uses Unsupervised Learning technique and clustering.

Applications:

1. Credit Card Fraud Detection.
2. The Phonetic Typewriter.
3. Customer Segmentation.

Generative Adversarial Network (GAN)

Generative Adversarial Networks (GANs) are a powerful class of neural networks that are used for unsupervised learning.

Architecture:

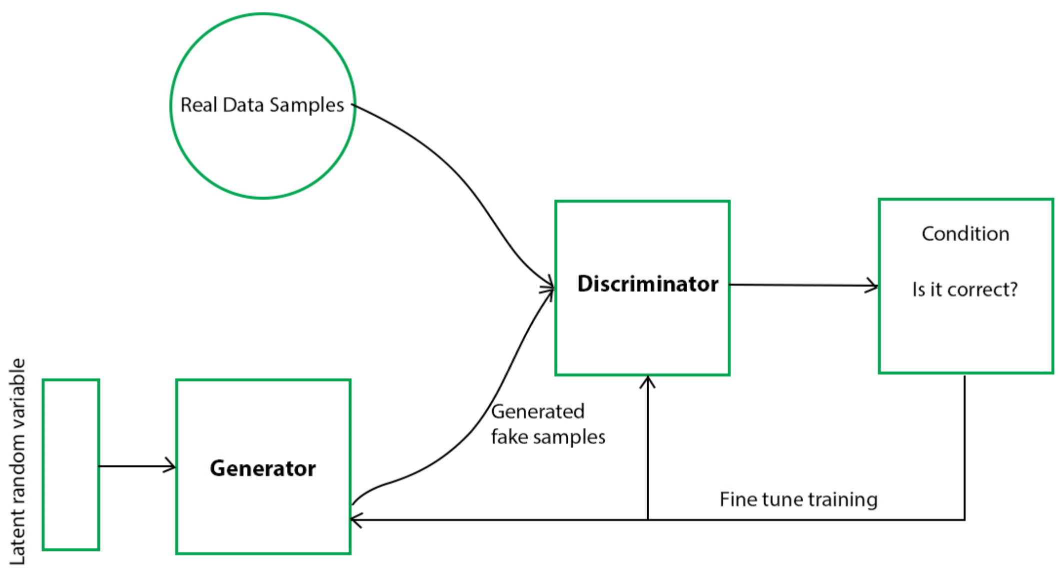


Figure 3: Architecture of GAN

Activation function:

ReLU: f(x)=max(0,x).

Applications:

1. Used to develop drugs for cancer, dermatological disorders, fibrosis, Parkinson's, Alzheimer's.
2. Used to improve augmented reality (AR).
3. Text to Image Translation. Etc.

Q2: Limitations of RNN.

Solution:

Recurrent Neural Networks use a backpropagation algorithm for training, it is commonly known as Back-propagation Through Time (BTT).

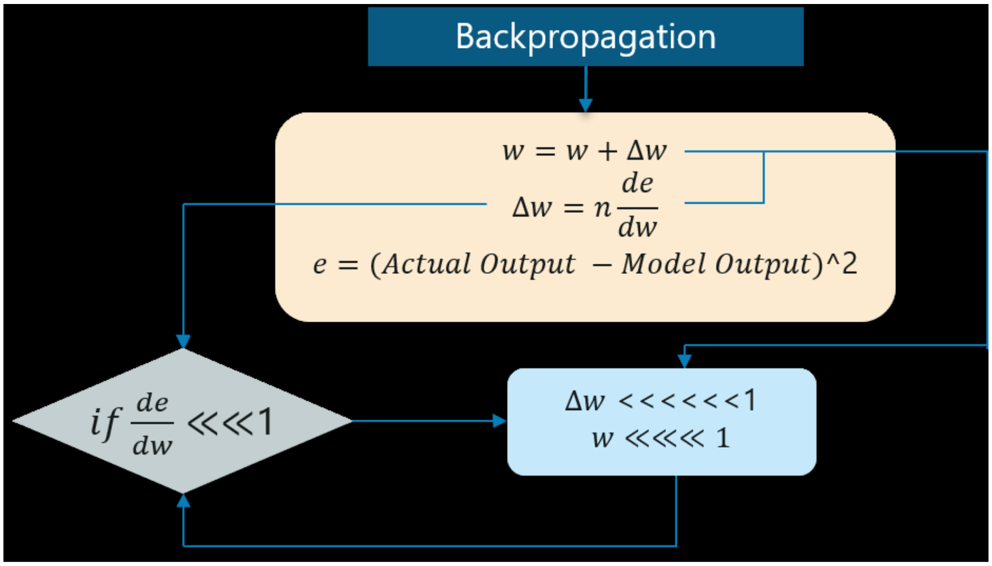
There are some issues with Back-propagation such as:

- Vanishing Gradient
- Exploding Gradient

Vanishing Gradient

When making use of back-propagation the goal is to calculate the error which is actually found out by finding out the difference between the actual output and the model output and raising that to a power of 2.

Consider the following diagram:

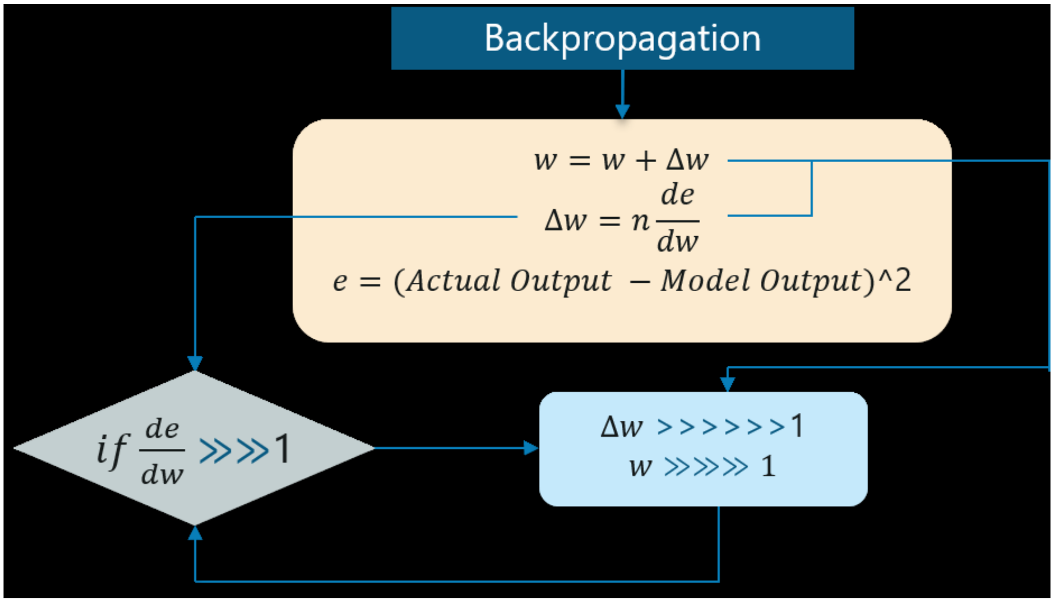


With the error calculated, the changes in the error with respect to the change in the weight is calculated. But with each learning rate, this has to be multiplied with the same. So, the product of the learning rate with the change leads to the value which is the actual change in the weight.

This change in weight is added to the old set of weights for every training iteration as shown in the figure. The issue here is when the change in weight is multiplied, the value is very less.

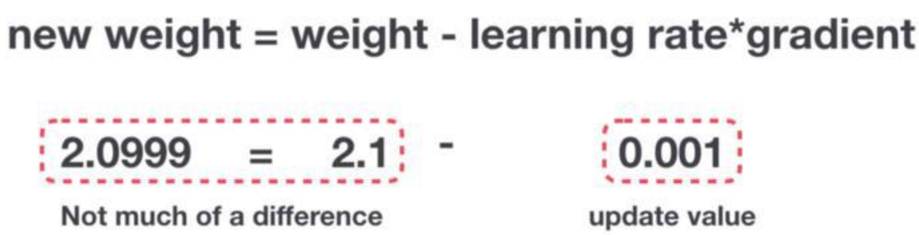
Exploding Gradient

The working of the exploding gradient is similar but the weights here change drastically instead of negligible change. Notice the small change in the diagram below:



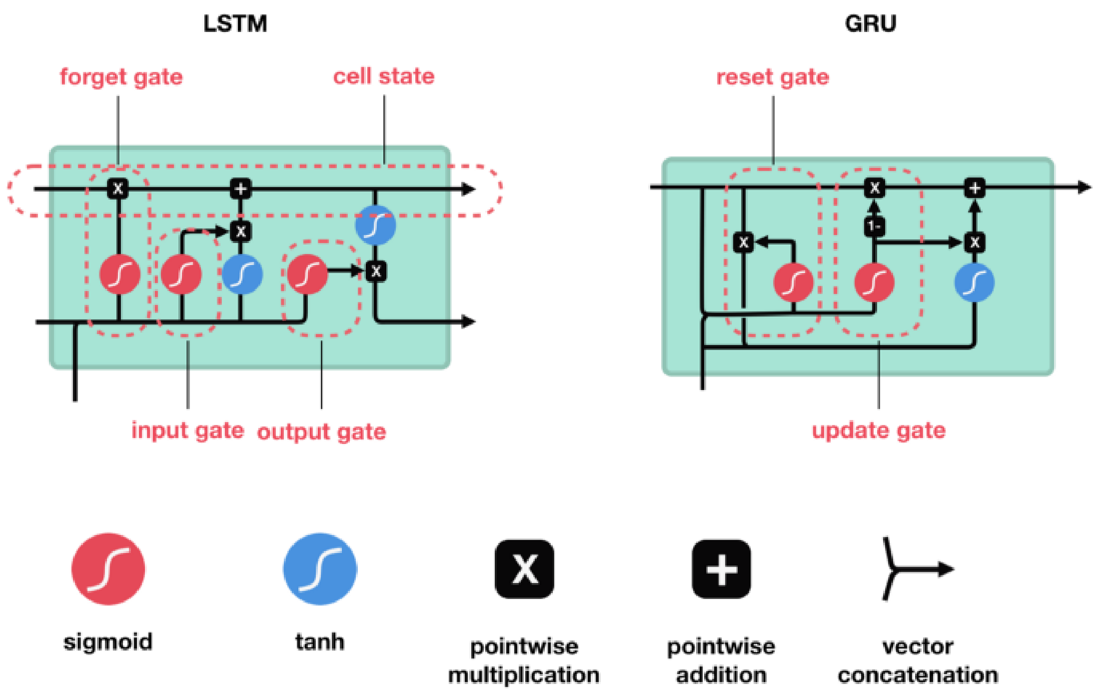
Q3: Limitations of RNN solve using LSTM and GRU (Or Only using LSTM).
Solution:

Recurrent Neural Networks suffer from short-term memory. If a sequence is long enough, they'll have a hard time carrying information from earlier time steps to later ones. So if we are trying to process a paragraph of text to do predictions, RNN's may leave out important information from the beginning. During backpropagation, recurrent neural networks suffer from the vanishing gradient problem. Gradients are values used to update the weights of a neural network. The vanishing gradient problem is when the gradient shrinks as it propagates through time. If a gradient value becomes extremely small, it doesn't contribute too much learning.



LSTM and GRU as a solution

LSTM 's and GRU's were created as the solution to short-term memory. They have internal mechanisms called gates that can regulate the flow of information.



These gates can learn which data in a sequence is important to keep or throw away. By doing that, it can pass relevant information down the long chain of sequences to make predictions. Almost all state of the art results based on recurrent neural networks are achieved with these two networks. LSTM and GRU's can be found in speech recognition, speech synthesis, and text generation. We can even use them to generate captions for videos.

Q4: The similarity and difference between Mobile net, google net and Inception network.
Solution:

Mobile Net-

Mobile Net is a type of convolutional neural network designed for mobile and embedded vision applications. It has 28 layers.

Google Net- Google Net is a type of CNN based on the inception architecture. Google Net
has 22 layers.

Inception- Inception is a CNN for identifying patterns in images. It allows multi types of filter size. It has 27 layers.

If we compare these three network, then we can see that among the three of them Mobile net has maximum number of layers in the architecture. So, the time complexity of mobile net is high but inception and google net has less time complexity because of less layers.

Q5: **How CNN works.**

Solution:

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion).

A convolution neural network has multiple hidden layers that help in extracting information from an image. The four important layers in CNN are:

1. Convolution layer
2. ReLU layer
3. Pooling layer
4. Fully connected layer

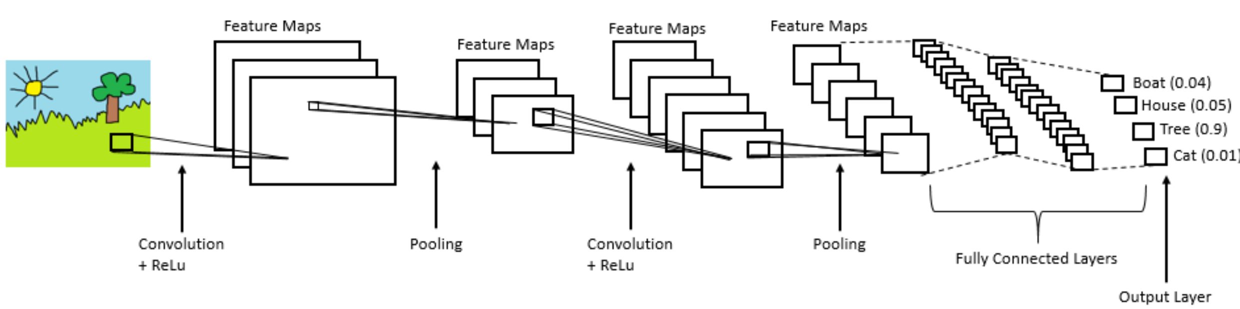


Figure: Complete CNN architecture

How it works:

- Provide input image into convolution layer
- Choose parameters, apply filters with strides, padding if requires. Perform convolution on the image and apply ReLU activation to the matrix.
- Perform pooling to reduce dimensionality size
- Add as many convolutional layers until satisfied
- Flatten the output and feed into a fully connected layer (FC Layer)
- Output the class using an activation function (Logistic Regression with cost functions) and classifies images.