Priyansh Singhal- 03815602717
WIBD Assignment 4

Q1 Explain why dropout in a neural network acts as a regularizer.

Deep neural nets with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks. Large networks are also slow to use, making it difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. Dropout is a technique for addressing this problem. The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. During training, dropout samples from an exponential number of different "thinned" networks. At test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights. This significantly reduces overfitting and gives major improvements over other regularization methods. We show that dropout improves the performance of neural networks on supervised learning tasks in vision, speech recognition, document classification and computational biology, obtaining state-of-the-art results on many benchmark data sets.
In a standard neural network, the derivative received by each parameter tells it how it should change so the final loss function is reduced, given what all other units are doing. Therefore, units may change in a way that they fix up the mistakes of the other units. This may lead to complex co-adaptations. This in turn leads to overfitting because these co-adaptations do not generalize to unseen data. We hypothesize that for each hidden unit, dropout prevents co-adaptation by making the presence of other hidden units unreliable. Therefore, a hidden unit cannot rely on other specific units to correct its mistakes. It must perform well in a wide variety of different contexts provided by the other hidden units. To observe this effect directly, we look at the first level features learned by neural networks trained on visual tasks with and without dropout.

Q2 Why is it necessary to introduce non-linearities in a neural network?

If we only allow linear activation functions in a neural network, the output will just be a linear transformation of the input, which is not enough to form a universal function approximator. Such a network can just be represented as a matrix multiplication, and you would not be able to obtain very interesting behaviors from such a network.
The same thing goes for the case where all neurons have affine activation functions (i.e. an activation function on the form f(x) = a*x + c, where a and c are constants, which is a generalization of linear activation functions), which will just result in an affine transformation from input to output, which is not very exciting either.
A neural network may very well contain neurons with linear activation functions, such as in the output layer, but these require the company of neurons with a non-linear activation function in other parts of the network.

Q3 What is exploding gradients ?

It can be understood as a recurrent neural network. For those who don't understand what a recurrent neural network is, can be intuited as a Neural network who gives feedback to its own self after every iteration of the self. Here feedback means the changing of the weight.
In gradient descent we try to find the global minimum of the cost function which will be the optimal solution for us.