**Assignment\_16**

1. Explain the Activation Functions in your own language

1. Sigmoid : **Sigmoid** **takes a real value as input and outputs another value between 0 and 1**.
2. Tanh **: Tanh** **squashes a real-valued number to the range [-1, 1]**. **It's non-linear. But unlike Sigmoid, its output is zero-centered.**
3. ReLU : **The rectified linear activation function or ReLU for short is** **a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero**. ELU
4. LeakyReLU : **Leaky Rectified Linear Unit, or Leaky ReLU**, i**s** **a type of activation function based on a ReLU, but it has a small slope for negative values instead of a flat slope**.
5. Swish : **Swish is** **a smooth, non-monotonic function that consistently matches or outperforms ReLU on deep networks applied to a variety of challenging domains such as Image classification and Machine translation**.

2. What happens when you increase or decrease the optimizer learning rate?

**Ans: he learning rate hyperparameter controls the rate or speed at which the model learns. Specifically, it controls the amount of apportioned error that the weights of the model are updated with each time they are updated, such as at the end of each batch of training examples.**

3. What happens when you increase the number of internal hidden neurons?

**Ans: An inordinately large number of neurons in the hidden layers can increase the time it takes to train the network. The amount of training time can increase to the point that it is impossible to adequately train the neural network.**

4. What happens when you increase the size of batch computation?

**Ans: Training loss and accuracy when the model is trained using different batch sizes. Testing loss and accuracy when the model is trained using different batch sizes. Finding: higher batch sizes leads to lower asymptotic test accuracy.**

5. Why we adopt regularization to avoid overfitting?

**Ans: Regularization comes into play and shrinks the learned estimates towards zero. In other words, it tunes the loss function by adding a penalty term, that prevents excessive fluctuation of the coefficients. Thereby, reducing the chances of overfitting**.

6. What are loss and cost functions in deep learning?

**Ans:** **Loss function: Used when we refer to the error for a single training example.** **Cost function: Used to refer to an average of the loss functions over an entire training dataset.**

**A cost function is a formula used to predict the cost that will be experienced at a certain activity level. This formula tends to be effective only within a range of activity levels, beyond which it no longer yields accurate results.**

7. What do you mean by underfitting in neural networks?

**Ans: A model is said to be underfitting when it's not able to classify the data it was trained on. We can tell that a model is underfitting when the metrics given for the training data are poor, meaning that the training accuracy of the model is low and/or the training loss is high.**

8. Why we use Dropout in Neural Networks?

**Ans: A Simple Way to Prevent Neural Networks from Overfitting. Dropout simulates a sparse activation from a given layer, which interestingly, in turn, encourages the network to actually learn a sparse representation as a side-effect.**