**MACHINE LEARNING WORKSHEET-1**

**Ans-1:** Least square Error

**Ans-2:** Linear Regression is sensitive to outliers

**Ans-3:** Negative

**Ans-4:** Both of them

**Ans-5:** Low bias and high variance

**Ans-6:** Predictive modal

**Ans-7:** Cross Validation

**Ans-8:** SMOTE

**Ans-9:** TPR and FPR

**Ans-10:** False

**Ans-11:** Apply PCA to project high dimensional data

**Ans-12:** We don’t have to choose the learning rate

It becomes slow when number of features is very large

**Ans-13: Regularization** refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting. Using Regularization, we can fit our machine learning model appropriately on a given test set and hence reduce the errors in it.

There are two main types of regularization techniques:

1. **Ridge Regularization**-Also known as Ridge Regression, it modifies the over-fitted or under fitted models by adding the penalty equivalent to the sum of the squares of the magnitude of coefficients.

This means that the mathematical function representing our machine learning model is minimized and coefficients are calculated. The magnitude of coefficients is squared and added. Ridge Regression performs regularization by shrinking the coefficients present.

2) **Lasso Regularization**.- It modifies the over-fitted or under-fitted models by adding the penalty equivalent to the sum of the absolute values of coefficients.

Lasso regression also performs coefficient minimization,but instead of squaring the magnitudes of the coefficients, it takes the true values of coefficients. This means that the coefficient sum can also be 0, because of the presence of negative coefficients.

**Ans-14:**

Ans-2

There are three main regularization techniques, namely:

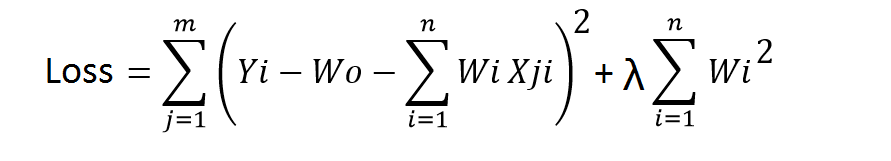
1. Ridge Regression (L2 Norm)
2. Lasso (L1 Norm)
3. Dropout

Ridge and Lasso can be used for any algorithms involving weight parameters, including neural nets. Dropout is primarily used in any kind of neural networks e.g. ANN, DNN, CNN or RNN to moderate the learning. Let’s take a closer look at each of the techniques.

**Ridge Regression (L2 Regularization)**

Ridge regression is also called L2 norm or regularization.

When using this technique, we add the sum of weight’s square to a loss function and thus create a new loss function which is denoted thus:



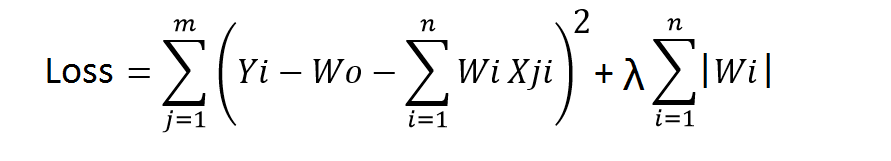
As seen above, the original loss function is modified by adding normalized weights. Here normalized weights are in the form of squares.

You may have noticed parameters λ along with normalized weights. λ is the parameter that needs to be tuned using a cross-validation dataset. When you use λ=0, it returns the residual sum of square as loss function which you chose initially. For a very high value of λ, loss will ignore core loss function and minimize weight’s square and will end up taking the parameters’ value as zero.

Now the parameters are learned using a modified loss function. To minimize the above function, parameters need to be as small as possible. Thus, L2 norm prevents weights from rising too high.

**Lasso Regression (L1 Regularization)**

Also called lasso regression and denoted as below:



This technique is different from ridge regression as it uses absolute weight values for normalization. λ is again a tuning parameter and behaves in the same as it does when using ridge regression.

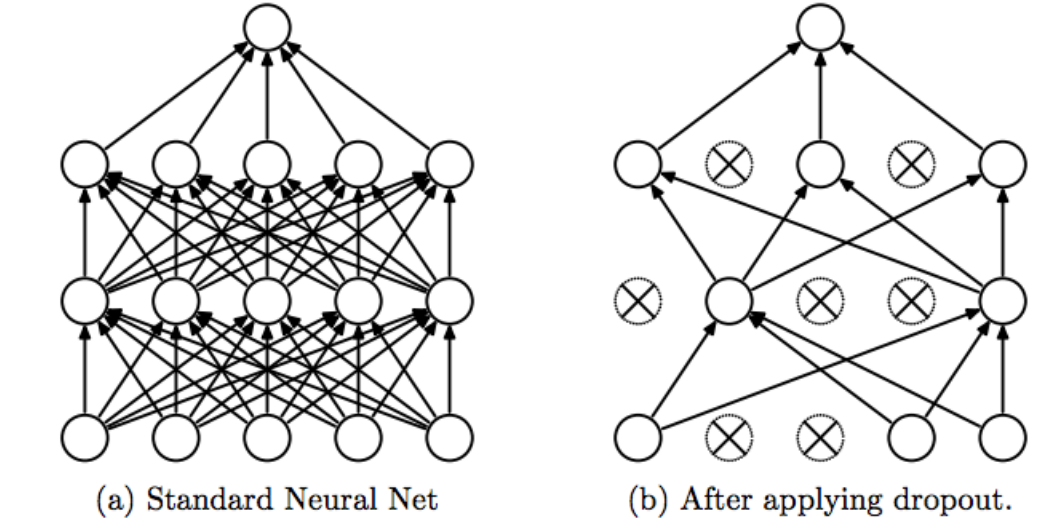
As loss function only considers absolute weights, optimization algorithms penalize higher weight values.

In ridge regression, loss function along with the optimization algorithm brings parameters near to zero but not actually zero, while lasso eliminates less important features and sets respective weight values to zero. Thus, lasso also performs feature selection along with regularization.

**Dropout**

Dropout is a regularization technique used in neural networks. It prevents complex co-adaptations from other neurons.

In neural nets, fully connected layers are more prone to overfit on training data. Using dropout, you can drop connections with *1-p* probability for each of the specified layers. Where *p* is called **keep probability** **parameter** and which needs to be tuned.



With dropout, you are left with a reduced network as dropped out neurons are left out during that training iteration.

Dropout decreases overfitting by avoiding training all the neurons on the complete training data in one go. It also improves training speed and learns more robust internal functions that generalize better on unseen data. However, it is important to note that Dropout takes more epochs to train compared to training without Dropout (If you have 10000 observations in your training data, then using 10000 examples for training is considered as 1 epoch).

Along with Dropout, neural networks can be regularized also using L1 and L2 norms. Apart from that, if you are working on an image dataset, [image augmentation](https://datamonje.com/image-data-augmentation/) can also be used as a regularization method.

For real-world applications, it is a must that a model performs well on unseen data. The techniques we discussed can help you make your model learn rather than just memorize.

Ans-3 [Linear regression](https://www.scribbr.com/statistics/simple-linear-regression/) most often uses mean-square error (MSE) to calculate the error of the model. MSE is calculated by:

1. measuring the distance of the observed y-values from the predicted y-values at each value of x;
2. squaring each of these distances;
3. calculating the[mean](https://www.scribbr.com/statistics/mean/) of each of the squared distances.

Linear regression fits a line to the data by finding the regression coefficient that results in the smallest MSE.