Disha D·N

Regularization in deep learning models.

→ Regularization is a set of techniques that can prevent overfitting in neural networks and thus improve the accuracy of a deep learning model

→ overfitting: It is the situation where the model works well for the training data but fails during testing.

→ overfitting is caused by noise in the training data that the neural network picks up during training and learns it an underlying concept of the data.

→ This learned noise, however, it is unique to each training set. As soon as the model sees new data from the same problem domain, but that does not contain this noise, the performance of the neural network gets much worse.

→ Less complex neural networks are less susceptible to overfitting. To prevent overfitting we use a technique called regularization.

→ Regularization: The techniques which helps to lower the the complexity of a neural network model during training and hence prevents overfitting.

→ There are three efficient regularization techniques namely L1, L2 and dropout.

L2 Regularization: It is also called as ridge regres

It prevents the overfitting by adding a penalty ter to the loss function.

$$\text{Loss} = (\hat{y} - y)$$

In L2 Loss,

$$\text{Loss(L2)} = \text{Loss} + \lambda * (\text{sum of squares of the weights})$$

where λ is called regularized paramet

→ The λ controls the strength of regularization. A larger λ value results in a greater penalty for larg weights, resulting in a more complex model that may fit the training data better but could overfit.

→ By adding the L2 regularization term, the model is encouraged to learn smaller and smoother weights, which can lead to improved generalization performan and better results on unseen data.

→ L2 regularization is particularly effective when there are many features on input data, as it helps to prevent the model from relying too heavily on any one feature.

→ But, the major disadvantage is, it is biased towards smaller weights and choosing the value for λ is a major challenge.

Regularisation: This is also called as Lasso Regression

→ It also adds a penalty to the loss function. The penalty term is the sum of the absolute values of the model's weights multiplied by a hyperparameter λ (lambda)

→ The L1 regularisation term is added to the standard loss function to obtain the regularised loss function

Regularized loss(L1) = loss * λ * (sum of absolute values of weights)

→ The λ parameter controls the strength of the regularisation. A larger value results in a greater penalty for larger weights, leading to a simple model with smaller coefficients.

Dropout in neural network.

Consider a neural network as shown in figure below

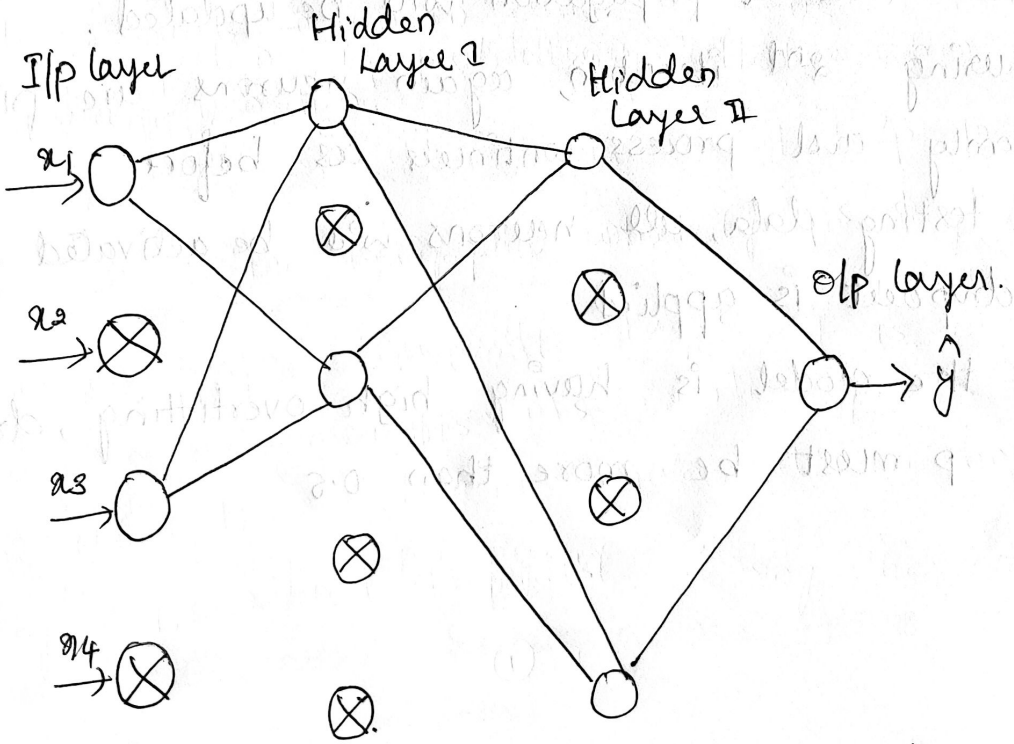


Fig: Neural network with dropout p = 0.5

→ Dropout refers to the practice of disregarding cer nodes in a layer at random during testing.

→ A dropout is also a regularization approach to prevent overfitting in deep neural networks

→ In drop out method, during training, some laye outputs are ignored or droped at random.

→ This makes the layer appear and is regarded as having a different number of nodes and connectedness to the preceding layer.

→ In drop out method, first we need to select a drop out ratio (p is in the range of 0 to 1)

→ Based on this ratio, subset of the features, hidden neurons are picked randomly during forward propagation. It makes the other neurons deactivated.

→ In backward propagation, whichever neurons activated in the forward propagation will be updated.

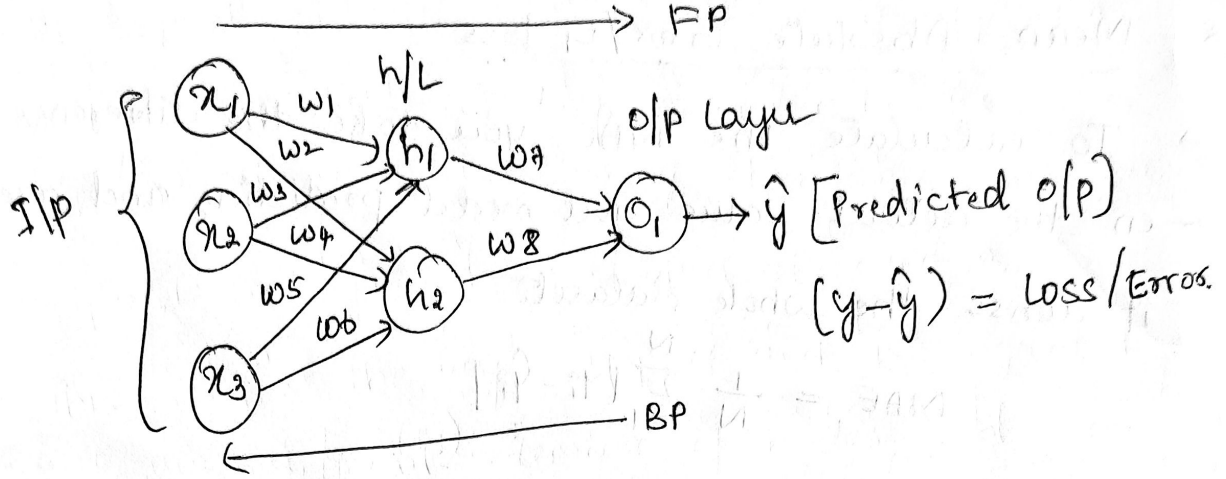
→ During 2nd iteration, again neurons are picked randomly and process continues as before.

→ In testing data, all neurons will be activated and no dropout is applied.

→ If the model is having high overfitting, dropout ratio p must be more than 0.5

Loss functions in neural network.

Consider a neural network,



→ Loss function is a method of evaluating how well your algorithm is modeling the given dataset.

→ Loss function helps to improve model's performance.

→ If the error is calculated for single training data then the error function is called as loss function, else it is called as cost function.

Loss functions in Regression.

① Mean squared Error / squared loss / L_2 loss (MSE)

→ To calculate the MSE, we need to take the actual value and model prediction, square it and average it across the whole dataset.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Advantages

1. Easy to interpret
2. Always differentiable because of the square
3. Only one local minima

Disadvantage:

1. Not robust to outliers

(2) Mean Absolute Error / L1 loss

→ To calculate the MAE, you take the difference bet-en the actual value and model prediction and averag if across the whole dataset.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Advantage

1. Intuitive and easy
2. Error unit same as the output column
3. Robust to outlier.

Disadvantage

1. Graph, not differential. we can not use gradient descent directly.

(3) Huber loss

→ It is a loss function used in robust regression, that is less sensitive to outliers in data than the squared error loss

$$\text{Huber} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(y_i - \hat{y}_i)^2 \qquad |y_i - \hat{y}_i| \le \delta$$

$$\text{Huber} = \frac{1}{n}\sum_{i=1}^{n}\delta(|y_i - \hat{y}_i| - \tfrac{1}{2}\delta) \qquad |y_i - \hat{y}_i| > \delta$$

n - the number of data points

y - the actual value of the data point

$\hat{y}$ - the predicted value of the data point. This value is returned by the model.

δ - defines the point where the huber loss function transitions from a quadratic to linear.

Advantage

1. Robust to outlier
2. It lies between MAE and MSE

Disadvantage:

1. Its main disadvantage is the associated complexity. In order to maximize the model accuracy, the hyperparameter δ will also need to be optimized which increase the training requirements.

classification loss

1. Binary cross Entropy / log loss.

→ It is used in binary classification problems like two classes (Eg: yes or No)

→ Binary cross entropy compares each of the predicted probabilities to the actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value.

$$\text{log loss} = -\frac{1}{n}\sum_{i=1}^{n} y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)$$

$y_i \rightarrow$ actual values

$\hat{y}_i \rightarrow$ Neural network predicted value

Advantage

1. Cost function is a differential

Disadvantage

1. Multiple local minima

(2) categorical Cross Entropy.

→ It is used for multiclass classification

→ It is also used in softmax regression

$$\text{loss} = -\sum_{j=1}^{K} y_j \log(\hat{y}_j)$$

where K ← No. of classes in the data

$$\text{Cost} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{K} [y_{ij} \log(\hat{y}_{ij})]$$

where K is number of classes

y is actual o/p value

$\hat{y}$ is the neural network prediction.