DEEP Learning:
Predicting bike sharing data

FUNCTIONS

- **Sigmoid** : Used as Activation function to introduce non-linearity after hidden layers and when we need outputs to be between 0 and 1.

$$S(x) = \frac{1}{1 + e^{-x}}$$

- **Mean Squared Error**: Used as a loss function when output is continuous.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

FORWARD PROPAGATION

- **Input layer** : Input preprocessed features.
- **Hidden Layer** : Matrix multiply with **weight(W)** matrices and add **Biases(B)** followed by activation function**(sigmoid)** for non-linearity.
- **Output Layer** : Matrix multiply with **weight(W)** matrices and add **Biases(B)** followed by activation function**(Identity)** based on required output.

DERIVATIVES

- **For Sigmoid:** Derivation of sigmoid function given above.

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

- **For normal layer operations:**

$$f(x) = Wx + B \implies f'(x) = W$$

- **Chain rule:**

$$\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$$

SETTING HYPERPARAMETERS

- **Iterations** :
  - Increase if both train and validation losses are decreasing.
  - Stop if validation loss has saturated.
  - Decrease if validation loss starts increasing.
- **Learning_rate** :
  - Increase if train loss is not changing or decreasing fast enough.
  - Decrease if train loss keeps fluctuating or starts increasing.
- **Hidden nodes** :
  - Increasing it makes it harder and longer for the model to train.
  - Increase when train loss saturates at a high value.
  - Decrease if train loss does not decreases very slowly or keeps fluctuating even after adjusting the learning_rate.

REFERENCES

- FeedForward explanation.
- Implementing backpropagation.
- Evaluation metrics