Predicting Deflection in a beam under stress

Deepankar(2018meb1219)
Jatin Gehlot(2018meb1231)
Saket Kumar(2018meb1260)

Introduction

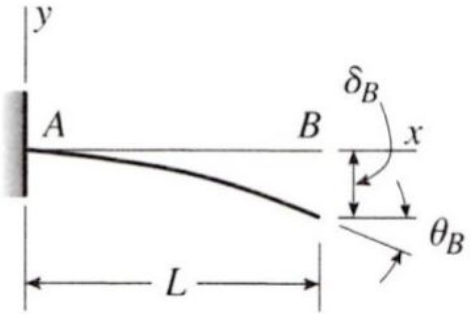
- In beams, under of force acting downward at distance ‘a’ from wall generate stress and give very small amount of deflection from starting position.
- Deflection and stress can be calculated by bending moment diagram.
- deflection and stress is function of x (as x is distance from wall) and a (distance of force from wall).
- This function can be obtain by follows governing equations-

$$M = EI\frac{d^2y}{dx^2} \quad \text{or} \quad \frac{M}{EI} = \frac{d^2y}{dx^2}$$

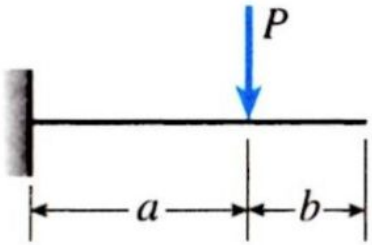
$$\sigma_{bend} = \frac{My}{I}$$

After making bending diagram for below figure we got moment function as-

$$M_{(x)} = \begin{cases} F(x-a), & for\ x < a \\ 0, & for\ x > a \end{cases}$$



v = deflection in the y direction (positive upward)
$v' = dv/dx$ = slope of the deflection curve
$\delta_B = -v(L)$ = deflection at end B of the beam (positive downward)
$\theta_B = -v'(L)$ = angle of rotation at end B of the beam (positive clockwise)
EI = constant



$$v = -\frac{Px^2}{6EI}(3a - x) \qquad v' = -\frac{Px}{2EI}(2a - x) \qquad (0 \le x \le a)$$

$$v = -\frac{Pa^2}{6EI}(3x - a) \qquad v' = -\frac{Pa^2}{2EI} \qquad (a \le x \le L)$$

Applications of machine learning in field-

- For constant cross-sectional area F/EI become constant for beam.
- This condition give advantage to predict the deflection by polynomial regression. As deflection remain function of x and a.
- We can apply neural network for better results from rough training data.

WORKFLOW AND APPROACH

- Data creation
- Model structure
- Performance evaluation
- Prediction

- **Data Creation**

  Created data for different values of x,a and calculated deflection using the given formula and stored these values in a csv file having 4 columns and 1000 samples(x,a,deflection and [x-a]).

$$\text{Deflection} = -\frac{F}{EI}\left(\frac{a\,x^2}{2} - \frac{x^3}{6} + \frac{\langle x-a\rangle^3}{6}\right)$$

$$\text{Moment} = F * ([x-a]) \qquad \text{where } [x-a] = \begin{cases} (x-a) , & \text{if } x<a \\ 0 , & \text{otherwise} \end{cases}$$

MODEL STRUCTURE

We used two approaches to predict with the highest accuracy possible

1. Predict the deflection using neural networks (Approach1)
2. Splitting the deflection formula in two parts and applying regression and NN to each parts. (Approach2)

$$\text{Deflection} = -\frac{F}{EI}\left(\frac{a\,x^2}{2} - \frac{x^3}{6} + \frac{\langle x-a\rangle^3}{6}\right)$$

Used polynomial regression

Used Neural Networks

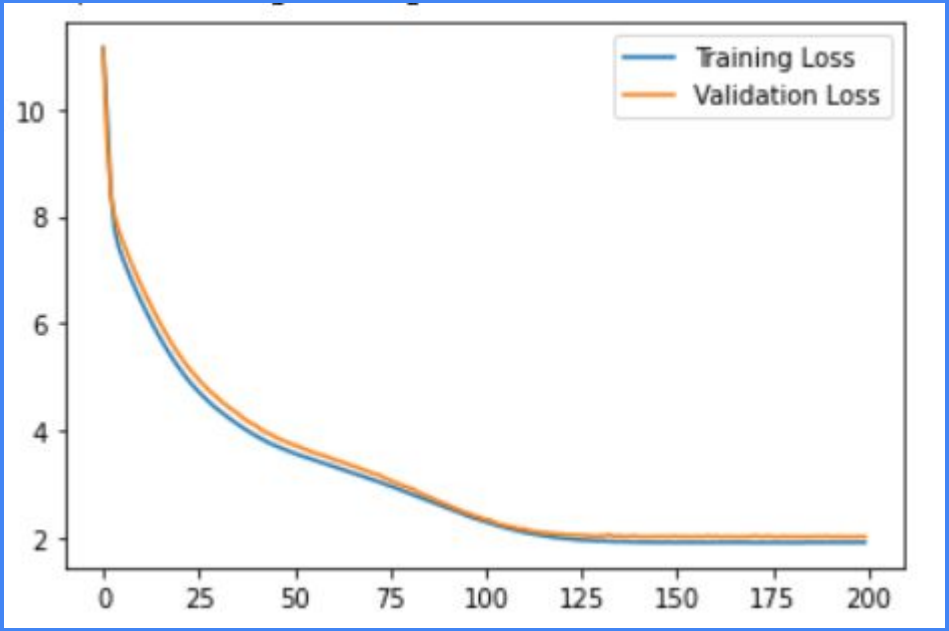
APPROACH 1

By just using Neural Network

- We have 2 input nodes (for x and a)
- Took 2 hidden layers
- Epoch- 200
- Batch size- 8
- Bias used in every layer
- Data splitted in training (80%) and test datas (20%)
- Optimizer - ADAM
- Activation Functions used - ReLU
- Loss - 4.05
- (Mean square error)- 2.0128
- Used tensor flow

Results of approach 1

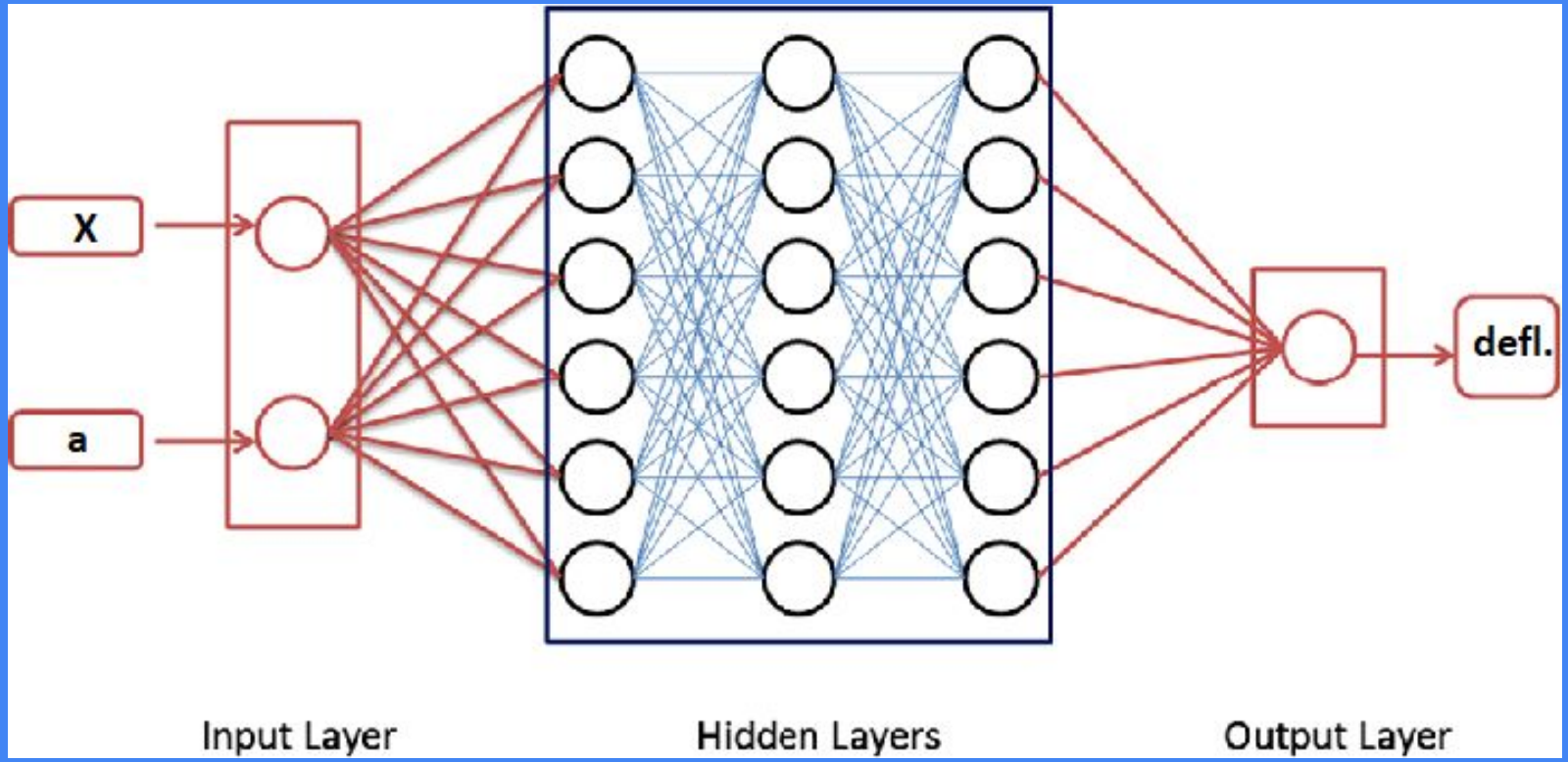
- Accuracy - 95%

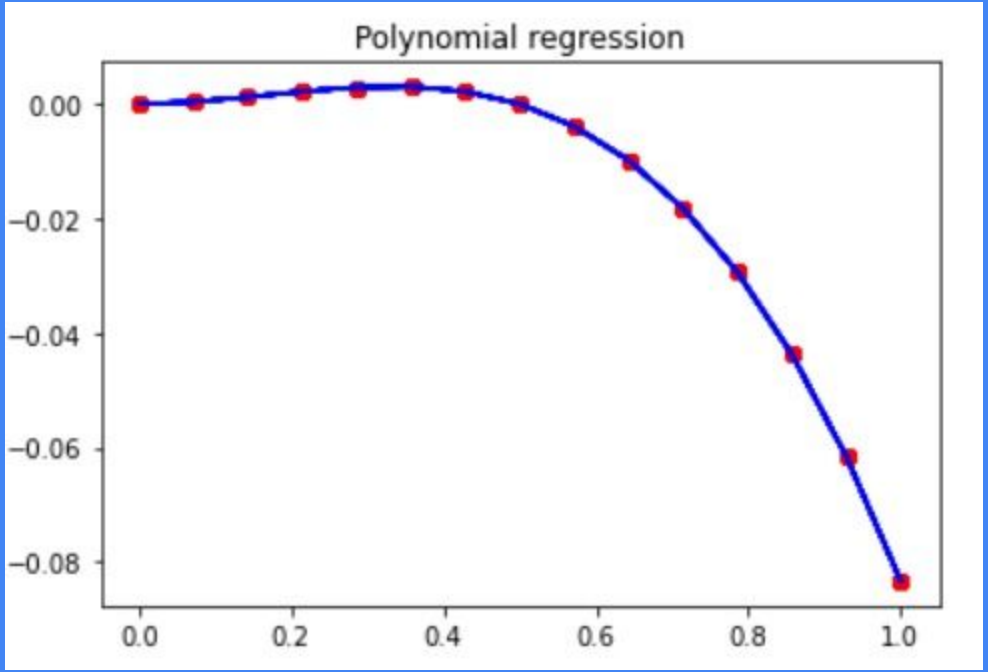


APPROACH 2

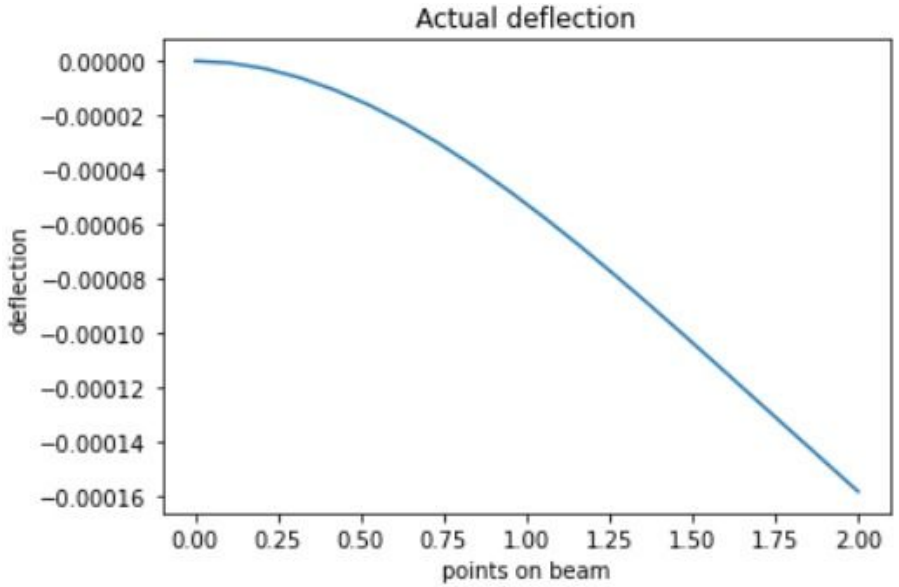
1. **Polynomial regression part**
   - Took one feature as a constant and then performed further.
   - degree= 3
   - We used sklearn for predicting the result for this part of the deflection.
   - Accuracy- 72.9%
2. **Neural Network part**
   - We took learning rate as 0.1 with 150 epochs
   - Optimization used - Gradient Descent
   - No bias used
   - Activation Function - Sigmoid
   - Loss - MSE (Mean Square error)- 0.6570
   - For using sigmoid function we reduced the input data to the values between 0 & 1 as it takes values in that range.
   - Accuracy - 88.6%

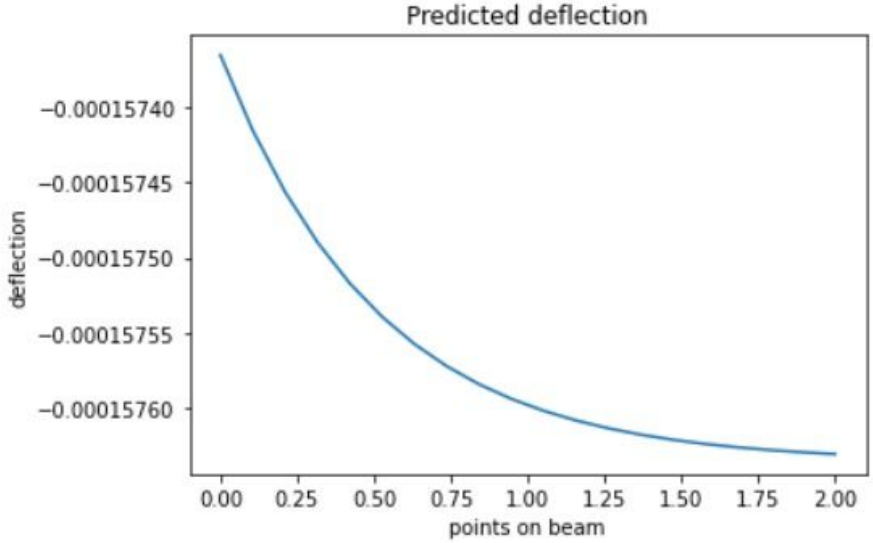
NN DIAGRAM




Polynomial regression
0.00
−0.02
−0.04
−0.06
−0.08
0.0
0.2
0.4
0.6
0.8
1.0

PERFORMANCE





Thanks!