Part 1:

Code:

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import matplotlib.pyplot as plt


# Define polynomial equations

def polynomial_equations(X):
    x1, x2, x3, x4, x5, x6, x7, x8 = X.T

    y1 = (2 * x1 * x3) - (x1 * x5) + (x3 * x8) + (2 * x1**2 * x8) + x5
    y2 = (x1 * x5 * x6) - (x3 * x4) - (3 * x2 * x3) + (2 * x2**2 * x4) - (2 * x
    y3 = (2 * x3**2) - (x5 * x7) - (3 * x1 * x4 * x6) + (x1**2 * x2 * x4) - 1
    y4 = (-x6**3) + (2.1 * x1 * x3 * x8) - (x1 * x4 * x7) - (3.2 * x5**2 * x2 *
    y5 = (x1**2 * x5) - (3 * x3 * x4 * x8) + (x1 * x2 * x4) - (3 * x6) + (x1**2

    return np.vstack([y1, y2, y3, y4, y5]).T



def add_noise(Y, noise_std=0.001, noise_ratio=0.3,seed=23):
    num_noisy_samples = int(Y.shape[0] * noise_ratio)
    np.random.seed(seed)
    noisy_indices = np.random.choice(Y.shape[0], num_noisy_samples, replace=Fal
    noise = np.zeros_like(Y)
    noise[noisy_indices] = np.random.normal(0, noise_std, Y.shape)[noisy_indice
    return Y + noise


def generate_data(N, noise_std=0.001, noise_ratio=0.1,seed=41):
    np.random.seed(seed)
    X = np.random.uniform(-1, 1, (N, 8))
    Y = polynomial_equations(X)
    Y = add_noise(Y, noise_std, noise_ratio)
    return X, Y
```

```
# Generate training and validation data
N_t, N_v = 1000, 500
X_train, Y_train = generate_data(N_t)
X_val, Y_val = generate_data(N_v, noise_std=0, noise_ratio=0,seed=53)  # No noi


def create_ffnn(hidden_layers=[6, 6, 6], activation_functions=['relu', 'tanh',
    model = Sequential()
    model.add(Dense(hidden_layers[0], activation=activation_functions[0], input
    model.add(Dense(hidden_layers[1], activation=activation_functions[1]))
    model.add(Dense(hidden_layers[2], activation=activation_functions[2]))
    model.add(Dense(5))  # Output layer updated to 5 neurons

    return model


def train_model(model, learning_rate=0.01, epochs=100,X_train=X_train,Y_train=Y
    model.compile(optimizer=keras.optimizers.SGD(learning_rate=learning_rate),
                  loss='mse', metrics=['mse'])

    history = model.fit(X_train, Y_train, validation_data=(X_val, Y_val), epoch
    return history


# Initialize Model and Train
model = create_ffnn()
history = train_model(model)

# Plot Training & Validation Loss
plt.plot(history.history['loss'], label='Train MSE')
plt.plot(history.history['val_loss'], label='Validation MSE')
plt.xlabel('Epochs')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

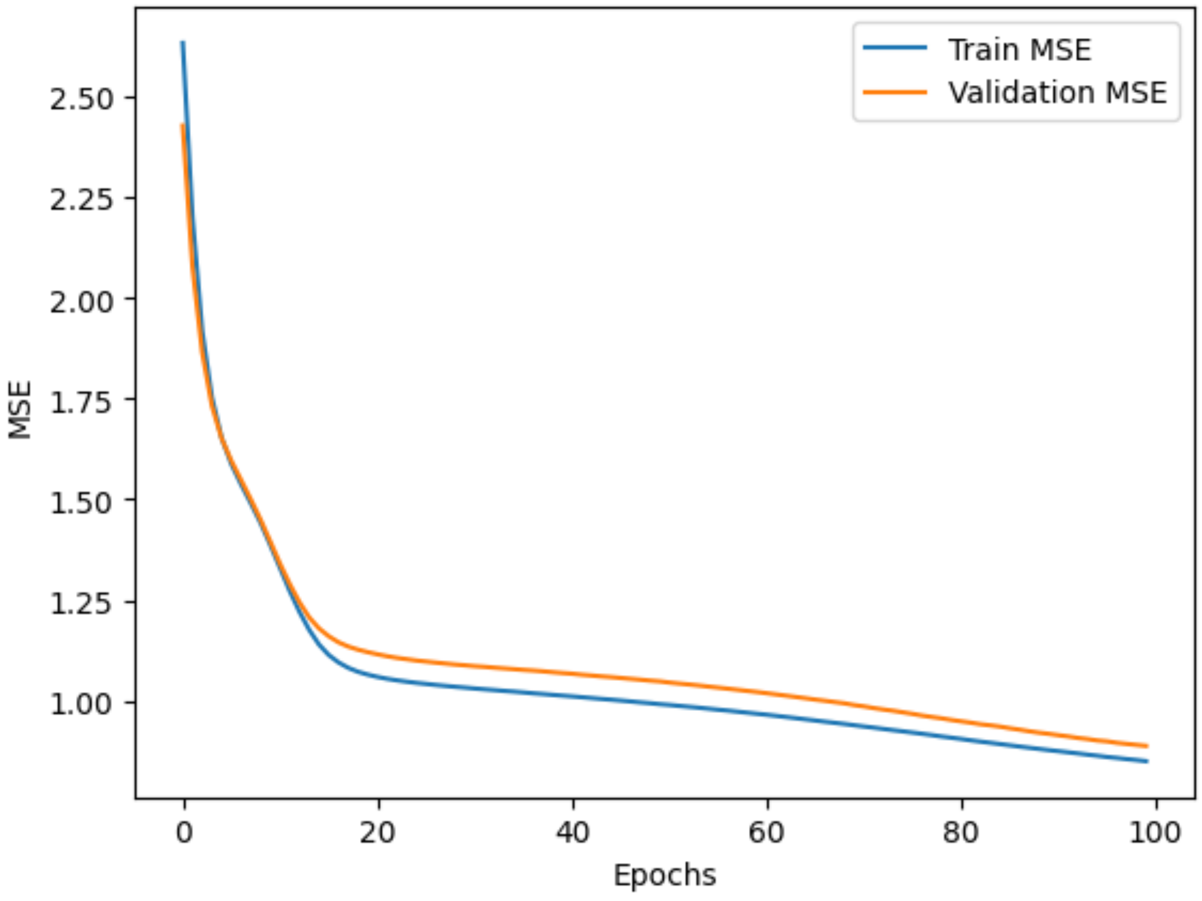
```
Epoch 1/100
32/32 - 2s - 48ms/step - loss: 2.6326 - mse: 2.6326 - val_loss: 2.4274 - va
Epoch 2/100
32/32 - 0s - 8ms/step - loss: 2.1997 - mse: 2.1997 - val_loss: 2.0763 - val
Epoch 3/100
32/32 - 0s - 12ms/step - loss: 1.9177 - mse: 1.9177 - val_loss: 1.8644 - va
Epoch 4/100
32/32 - 1s - 20ms/step - loss: 1.7531 - mse: 1.7531 - val_loss: 1.7319 - va
Epoch 5/100
32/32 - 0s - 14ms/step - loss: 1.6522 - mse: 1.6522 - val_loss: 1.6512 - va
Epoch 6/100
32/32 - 0s - 10ms/step - loss: 1.5871 - mse: 1.5871 - val_loss: 1.5953 - va
Epoch 7/100
32/32 - 0s - 9ms/step - loss: 1.5371 - mse: 1.5371 - val_loss: 1.5477 - val
```

```
Epoch 8/100
32/32 - 0s - 5ms/step - loss: 1.4901 - mse: 1.4901 - val_loss: 1.4998 - val
Epoch 9/100
32/32 - 0s - 10ms/step - loss: 1.4415 - mse: 1.4415 - val_loss: 1.4494 - va
Epoch 10/100
32/32 - 0s - 9ms/step - loss: 1.3868 - mse: 1.3868 - val_loss: 1.3951 - val
Epoch 11/100
32/32 - 0s - 9ms/step - loss: 1.3297 - mse: 1.3297 - val_loss: 1.3402 - val
Epoch 12/100
32/32 - 0s - 10ms/step - loss: 1.2725 - mse: 1.2725 - val_loss: 1.2896 - va
Epoch 13/100
32/32 - 0s - 9ms/step - loss: 1.2209 - mse: 1.2209 - val_loss: 1.2444 - val
Epoch 14/100
32/32 - 0s - 10ms/step - loss: 1.1761 - mse: 1.1761 - val_loss: 1.2072 - va
Epoch 15/100
32/32 - 0s - 9ms/step - loss: 1.1400 - mse: 1.1400 - val_loss: 1.1799 - val
Epoch 16/100
32/32 - 0s - 9ms/step - loss: 1.1140 - mse: 1.1140 - val_loss: 1.1605 - val
Epoch 17/100
32/32 - 0s - 9ms/step - loss: 1.0957 - mse: 1.0957 - val_loss: 1.1455 - val
Epoch 18/100
32/32 - 0s - 5ms/step - loss: 1.0822 - mse: 1.0822 - val_loss: 1.1350 - val
Epoch 19/100
32/32 - 0s - 10ms/step - loss: 1.0724 - mse: 1.0724 - val_loss: 1.1273 - va
Epoch 20/100
32/32 - 0s - 9ms/step - loss: 1.0653 - mse: 1.0653 - val_loss: 1.1211 - val
Epoch 21/100
32/32 - 0s - 6ms/step - loss: 1.0596 - mse: 1.0596 - val_loss: 1.1157 - val
Epoch 22/100
32/32 - 0s - 9ms/step - loss: 1.0549 - mse: 1.0549 - val_loss: 1.1109 - val
Epoch 23/100
32/32 - 0s - 6ms/step - loss: 1.0511 - mse: 1.0511 - val_loss: 1.1068 - val
Epoch 24/100
32/32 - 0s - 9ms/step - loss: 1.0477 - mse: 1.0477 - val_loss: 1.1036 - val
Epoch 25/100
32/32 - 0s - 9ms/step - loss: 1.0448 - mse: 1.0448 - val_loss: 1.1002 - val
Epoch 26/100
32/32 - 0s - 10ms/step - loss: 1.0422 - mse: 1.0422 - val_loss: 1.0977 - va
Epoch 27/100
32/32 - 0s - 5ms/step - loss: 1.0396 - mse: 1.0396 - val_loss: 1.0950 - val
Epoch 28/100
32/32 - 0s - 9ms/step - loss: 1.0372 - mse: 1.0372 - val_loss: 1.0927 - val
Epoch 29/100
32/32 - 0s - 9ms/step - loss: 1.0350 - mse: 1.0350 - val_loss: 1.0903 - val
Epoch 30/100
32/32 - 0s - 10ms/step - loss: 1.0330 - mse: 1.0330 - val_loss: 1.0883 - va
Epoch 31/100
32/32 - 0s - 6ms/step - loss: 1.0308 - mse: 1.0308 - val_loss: 1.0862 - val
Epoch 32/100
32/32 - 0s - 9ms/step - loss: 1.0288 - mse: 1.0288 - val_loss: 1.0845 - val
Epoch 33/100
32/32 - 0s - 5ms/step - loss: 1.0269 - mse: 1.0269 - val_loss: 1.0825 - val
Epoch 34/100
32/32 - 0s - 5ms/step - loss: 1.0249 - mse: 1.0249 - val_loss: 1.0807 - val
```

```
Epoch 35/100
32/32 - 0s - 5ms/step - loss: 1.0232 - mse: 1.0232 - val_loss: 1.0789 - val
Epoch 36/100
32/32 - 0s - 9ms/step - loss: 1.0211 - mse: 1.0211 - val_loss: 1.0769 - val
Epoch 37/100
32/32 - 0s - 9ms/step - loss: 1.0189 - mse: 1.0189 - val_loss: 1.0754 - val
Epoch 38/100
32/32 - 0s - 10ms/step - loss: 1.0171 - mse: 1.0171 - val_loss: 1.0734 - va
Epoch 39/100
32/32 - 0s - 9ms/step - loss: 1.0150 - mse: 1.0150 - val_loss: 1.0713 - val
Epoch 40/100
32/32 - 0s - 5ms/step - loss: 1.0131 - mse: 1.0131 - val_loss: 1.0692 - val
Epoch 41/100
32/32 - 0s - 6ms/step - loss: 1.0112 - mse: 1.0112 - val_loss: 1.0675 - val
Epoch 42/100
32/32 - 0s - 14ms/step - loss: 1.0092 - mse: 1.0092 - val_loss: 1.0650 - va
Epoch 43/100
32/32 - 1s - 19ms/step - loss: 1.0071 - mse: 1.0071 - val_loss: 1.0631 - va
Epoch 44/100
32/32 - 1s - 18ms/step - loss: 1.0050 - mse: 1.0050 - val_loss: 1.0609 - va
Epoch 45/100
32/32 - 0s - 11ms/step - loss: 1.0031 - mse: 1.0031 - val_loss: 1.0591 - va
Epoch 46/100
32/32 - 0s - 14ms/step - loss: 1.0009 - mse: 1.0009 - val_loss: 1.0570 - va
Epoch 47/100
32/32 - 0s - 10ms/step - loss: 0.9987 - mse: 0.9987 - val_loss: 1.0548 - va
Epoch 48/100
32/32 - 0s - 9ms/step - loss: 0.9964 - mse: 0.9964 - val_loss: 1.0527 - val
Epoch 49/100
32/32 - 0s - 9ms/step - loss: 0.9943 - mse: 0.9943 - val_loss: 1.0504 - val
Epoch 50/100
32/32 - 0s - 10ms/step - loss: 0.9919 - mse: 0.9919 - val_loss: 1.0485 - va
Epoch 51/100
32/32 - 0s - 9ms/step - loss: 0.9900 - mse: 0.9900 - val_loss: 1.0456 - val
Epoch 52/100
32/32 - 0s - 6ms/step - loss: 0.9877 - mse: 0.9877 - val_loss: 1.0431 - val
Epoch 53/100
32/32 - 0s - 6ms/step - loss: 0.9855 - mse: 0.9855 - val_loss: 1.0408 - val
Epoch 54/100
32/32 - 0s - 5ms/step - loss: 0.9832 - mse: 0.9832 - val_loss: 1.0383 - val
Epoch 55/100
32/32 - 0s - 9ms/step - loss: 0.9808 - mse: 0.9808 - val_loss: 1.0356 - val
Epoch 56/100
32/32 - 0s - 5ms/step - loss: 0.9783 - mse: 0.9783 - val_loss: 1.0330 - val
Epoch 57/100
32/32 - 0s - 10ms/step - loss: 0.9760 - mse: 0.9760 - val_loss: 1.0303 - va
Epoch 58/100
32/32 - 0s - 9ms/step - loss: 0.9733 - mse: 0.9733 - val_loss: 1.0276 - val
Epoch 59/100
32/32 - 0s - 9ms/step - loss: 0.9708 - mse: 0.9708 - val_loss: 1.0249 - val
Epoch 60/100
32/32 - 0s - 5ms/step - loss: 0.9682 - mse: 0.9682 - val_loss: 1.0221 - val
Epoch 61/100
32/32 - 0s - 10ms/step - loss: 0.9657 - mse: 0.9657 - val_loss: 1.0191 - va
```

```
Epoch 62/100
32/32 - 0s - 9ms/step - loss: 0.9629 - mse: 0.9629 - val_loss: 1.0161 - val
Epoch 63/100
32/32 - 0s - 5ms/step - loss: 0.9601 - mse: 0.9601 - val_loss: 1.0130 - val
Epoch 64/100
32/32 - 0s - 5ms/step - loss: 0.9572 - mse: 0.9572 - val_loss: 1.0098 - val
Epoch 65/100
32/32 - 0s - 9ms/step - loss: 0.9543 - mse: 0.9543 - val_loss: 1.0066 - val
Epoch 66/100
32/32 - 0s - 5ms/step - loss: 0.9514 - mse: 0.9514 - val_loss: 1.0033 - val
Epoch 67/100
32/32 - 0s - 6ms/step - loss: 0.9484 - mse: 0.9484 - val_loss: 1.0003 - val
Epoch 68/100
32/32 - 0s - 9ms/step - loss: 0.9456 - mse: 0.9456 - val_loss: 0.9971 - val
Epoch 69/100
32/32 - 0s - 9ms/step - loss: 0.9428 - mse: 0.9428 - val_loss: 0.9940 - val
Epoch 70/100
32/32 - 0s - 10ms/step - loss: 0.9396 - mse: 0.9396 - val_loss: 0.9897 - va
Epoch 71/100
32/32 - 0s - 9ms/step - loss: 0.9365 - mse: 0.9365 - val_loss: 0.9864 - val
Epoch 72/100
32/32 - 0s - 10ms/step - loss: 0.9337 - mse: 0.9337 - val_loss: 0.9823 - va
Epoch 73/100
32/32 - 0s - 9ms/step - loss: 0.9305 - mse: 0.9305 - val_loss: 0.9787 - val
Epoch 74/100
32/32 - 0s - 5ms/step - loss: 0.9274 - mse: 0.9274 - val_loss: 0.9755 - val
Epoch 75/100
32/32 - 0s - 6ms/step - loss: 0.9245 - mse: 0.9245 - val_loss: 0.9717 - val
Epoch 76/100
32/32 - 0s - 5ms/step - loss: 0.9214 - mse: 0.9214 - val_loss: 0.9678 - val
Epoch 77/100
32/32 - 0s - 5ms/step - loss: 0.9184 - mse: 0.9184 - val_loss: 0.9635 - val
Epoch 78/100
32/32 - 0s - 9ms/step - loss: 0.9152 - mse: 0.9152 - val_loss: 0.9601 - val
Epoch 79/100
32/32 - 0s - 5ms/step - loss: 0.9117 - mse: 0.9117 - val_loss: 0.9567 - val
Epoch 80/100
32/32 - 0s - 6ms/step - loss: 0.9087 - mse: 0.9087 - val_loss: 0.9527 - val
Epoch 81/100
32/32 - 0s - 9ms/step - loss: 0.9056 - mse: 0.9056 - val_loss: 0.9493 - val
Epoch 82/100
32/32 - 0s - 9ms/step - loss: 0.9021 - mse: 0.9021 - val_loss: 0.9459 - val
Epoch 83/100
32/32 - 0s - 10ms/step - loss: 0.8990 - mse: 0.8990 - val_loss: 0.9421 - va
Epoch 84/100
32/32 - 0s - 6ms/step - loss: 0.8960 - mse: 0.8960 - val_loss: 0.9394 - val
Epoch 85/100
32/32 - 0s - 14ms/step - loss: 0.8929 - mse: 0.8929 - val_loss: 0.9362 - va
Epoch 86/100
32/32 - 1s - 17ms/step - loss: 0.8898 - mse: 0.8898 - val_loss: 0.9321 - va
Epoch 87/100
32/32 - 0s - 9ms/step - loss: 0.8867 - mse: 0.8867 - val_loss: 0.9286 - val
Epoch 88/100
32/32 - 0s - 10ms/step - loss: 0.8838 - mse: 0.8838 - val_loss: 0.9247 - va
```

```
Epoch 89/100
32/32 - 0s - 11ms/step - loss: 0.8809 - mse: 0.8809 - val_loss: 0.9211 - va
Epoch 90/100
32/32 - 0s - 15ms/step - loss: 0.8779 - mse: 0.8779 - val_loss: 0.9183 - va
Epoch 91/100
32/32 - 0s - 5ms/step - loss: 0.8753 - mse: 0.8753 - val_loss: 0.9149 - val
Epoch 92/100
32/32 - 0s - 9ms/step - loss: 0.8724 - mse: 0.8724 - val_loss: 0.9118 - val
Epoch 93/100
32/32 - 0s - 9ms/step - loss: 0.8697 - mse: 0.8697 - val_loss: 0.9081 - val
Epoch 94/100
32/32 - 0s - 10ms/step - loss: 0.8667 - mse: 0.8667 - val_loss: 0.9053 - va
Epoch 95/100
32/32 - 0s - 9ms/step - loss: 0.8639 - mse: 0.8639 - val_loss: 0.9017 - val
Epoch 96/100
32/32 - 0s - 9ms/step - loss: 0.8608 - mse: 0.8608 - val_loss: 0.8996 - val
Epoch 97/100
32/32 - 0s - 6ms/step - loss: 0.8585 - mse: 0.8585 - val_loss: 0.8960 - val
Epoch 98/100
32/32 - 0s - 9ms/step - loss: 0.8556 - mse: 0.8556 - val_loss: 0.8933 - val
Epoch 99/100
32/32 - 0s - 10ms/step - loss: 0.8533 - mse: 0.8533 - val_loss: 0.8909 - va
Epoch 100/100
32/32 - 0s - 9ms/step - loss: 0.8505 - mse: 0.8505 - val_loss: 0.8884 - val
```



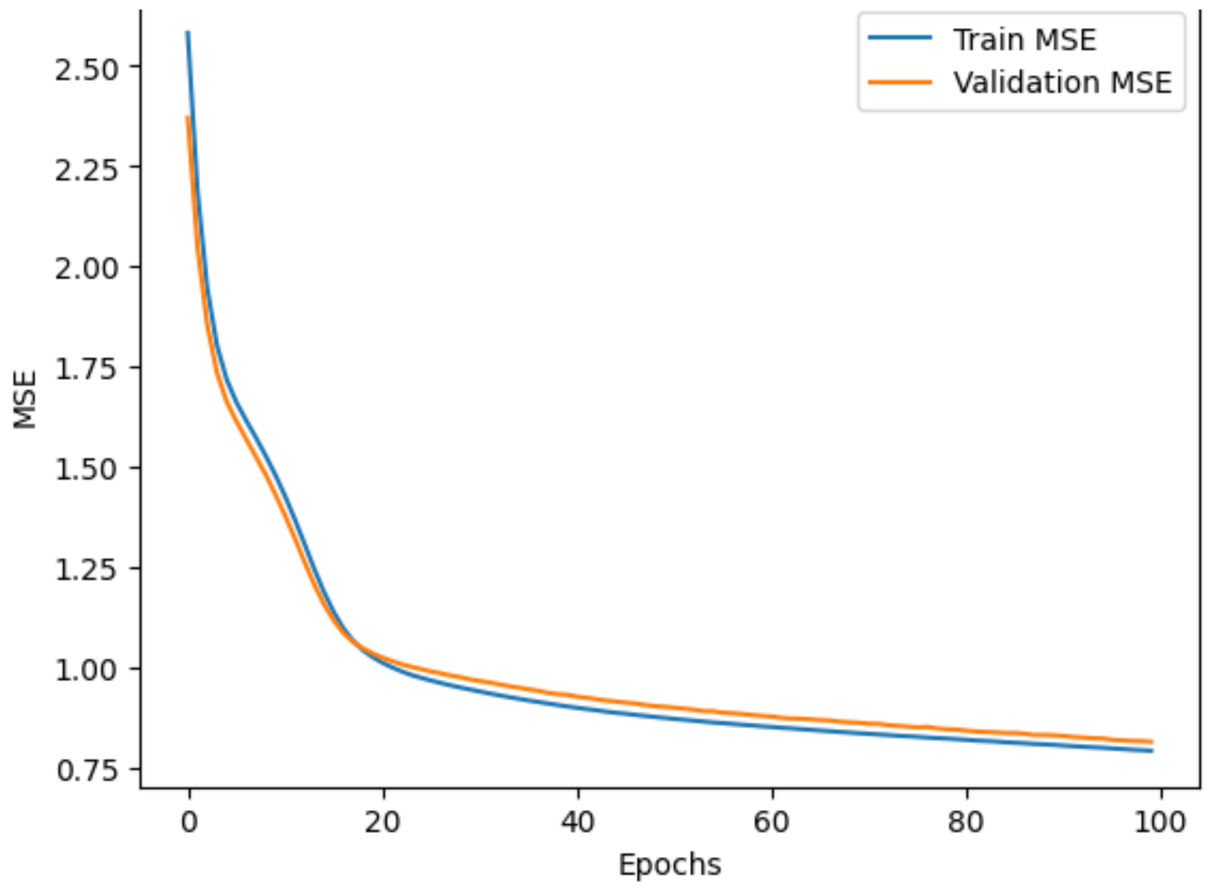
```
# Model With Different Activation Functions - 1
model_2 = create_ffnn(activation_functions=['tanh', 'leaky_relu', 'linear'])
history_2 = train_model(model_2)

# Plot Training & Validation Loss
plt.plot(history_2.history['loss'], label='Train MSE')
plt.plot(history_2.history['val_loss'], label='Validation MSE')
plt.xlabel('Epochs')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

```
Epoch 1/100
32/32 - 1s - 39ms/step - loss: 2.5815 - mse: 2.5815 - val_loss: 2.3696 - va
Epoch 2/100
32/32 - 0s - 11ms/step - loss: 2.1878 - mse: 2.1878 - val_loss: 2.0469 - va
Epoch 3/100
32/32 - 1s - 20ms/step - loss: 1.9454 - mse: 1.9454 - val_loss: 1.8536 - va
Epoch 4/100
32/32 - 0s - 15ms/step - loss: 1.8026 - mse: 1.8026 - val_loss: 1.7343 - va
Epoch 5/100
32/32 - 0s - 9ms/step - loss: 1.7172 - mse: 1.7172 - val_loss: 1.6617 - val
Epoch 6/100
32/32 - 0s - 9ms/step - loss: 1.6604 - mse: 1.6604 - val_loss: 1.6127 - val
Epoch 7/100
32/32 - 0s - 9ms/step - loss: 1.6154 - mse: 1.6154 - val_loss: 1.5687 - val
Epoch 8/100
32/32 - 0s - 9ms/step - loss: 1.5727 - mse: 1.5727 - val_loss: 1.5245 - val
Epoch 9/100
32/32 - 0s - 5ms/step - loss: 1.5275 - mse: 1.5275 - val_loss: 1.4807 - val
Epoch 10/100
32/32 - 0s - 9ms/step - loss: 1.4795 - mse: 1.4795 - val_loss: 1.4307 - val
Epoch 11/100
32/32 - 0s - 10ms/step - loss: 1.4267 - mse: 1.4267 - val_loss: 1.3781 - va
Epoch 12/100
32/32 - 0s - 10ms/step - loss: 1.3686 - mse: 1.3686 - val_loss: 1.3213 - va
Epoch 13/100
32/32 - 0s - 5ms/step - loss: 1.3076 - mse: 1.3076 - val_loss: 1.2622 - val
Epoch 14/100
32/32 - 0s - 9ms/step - loss: 1.2449 - mse: 1.2449 - val_loss: 1.2061 - val
Epoch 15/100
32/32 - 0s - 9ms/step - loss: 1.1872 - mse: 1.1872 - val_loss: 1.1564 - val
Epoch 16/100
32/32 - 0s - 5ms/step - loss: 1.1374 - mse: 1.1374 - val_loss: 1.1167 - val
Epoch 17/100
32/32 - 0s - 10ms/step - loss: 1.0963 - mse: 1.0963 - val_loss: 1.0847 - va
Epoch 18/100
32/32 - 0s - 10ms/step - loss: 1.0652 - mse: 1.0652 - val_loss: 1.0627 - va
Epoch 19/100
32/32 - 0s - 9ms/step - loss: 1.0424 - mse: 1.0424 - val_loss: 1.0468 - val
Epoch 20/100
```

```
32/32 - 0s - 10ms/step - loss: 1.0249 - mse: 1.0249 - val_loss: 1.0343 - va
Epoch 21/100
32/32 - 0s - 5ms/step - loss: 1.0110 - mse: 1.0110 - val_loss: 1.0236 - val
Epoch 22/100
32/32 - 0s - 10ms/step - loss: 0.9994 - mse: 0.9994 - val_loss: 1.0149 - va
Epoch 23/100
32/32 - 0s - 9ms/step - loss: 0.9892 - mse: 0.9892 - val_loss: 1.0070 - val
Epoch 24/100
32/32 - 0s - 10ms/step - loss: 0.9801 - mse: 0.9801 - val_loss: 1.0008 - va
Epoch 25/100
32/32 - 0s - 9ms/step - loss: 0.9732 - mse: 0.9732 - val_loss: 0.9950 - val
Epoch 26/100
32/32 - 0s - 9ms/step - loss: 0.9665 - mse: 0.9665 - val_loss: 0.9892 - val
Epoch 27/100
32/32 - 0s - 10ms/step - loss: 0.9604 - mse: 0.9604 - val_loss: 0.9845 - va
Epoch 28/100
32/32 - 0s - 5ms/step - loss: 0.9546 - mse: 0.9546 - val_loss: 0.9791 - val
Epoch 29/100
32/32 - 0s - 9ms/step - loss: 0.9492 - mse: 0.9492 - val_loss: 0.9749 - val
Epoch 30/100
32/32 - 0s - 9ms/step - loss: 0.9442 - mse: 0.9442 - val_loss: 0.9694 - val
Epoch 31/100
32/32 - 0s - 10ms/step - loss: 0.9394 - mse: 0.9394 - val_loss: 0.9653 - va
Epoch 32/100
32/32 - 0s - 9ms/step - loss: 0.9344 - mse: 0.9344 - val_loss: 0.9617 - val
Epoch 33/100
32/32 - 0s - 9ms/step - loss: 0.9297 - mse: 0.9297 - val_loss: 0.9571 - val
Epoch 34/100
32/32 - 0s - 10ms/step - loss: 0.9253 - mse: 0.9253 - val_loss: 0.9526 - va
Epoch 35/100
32/32 - 0s - 9ms/step - loss: 0.9211 - mse: 0.9211 - val_loss: 0.9487 - val
Epoch 36/100
32/32 - 0s - 9ms/step - loss: 0.9169 - mse: 0.9169 - val_loss: 0.9448 - val
Epoch 37/100
32/32 - 0s - 5ms/step - loss: 0.9133 - mse: 0.9133 - val_loss: 0.9410 - val
Epoch 38/100
32/32 - 0s - 10ms/step - loss: 0.9093 - mse: 0.9093 - val_loss: 0.9360 - va
Epoch 39/100
32/32 - 0s - 10ms/step - loss: 0.9056 - mse: 0.9056 - val_loss: 0.9324 - va
Epoch 40/100
32/32 - 0s - 9ms/step - loss: 0.9018 - mse: 0.9018 - val_loss: 0.9307 - val
Epoch 41/100
32/32 - 0s - 9ms/step - loss: 0.8986 - mse: 0.8986 - val_loss: 0.9264 - val
Epoch 42/100
32/32 - 0s - 10ms/step - loss: 0.8953 - mse: 0.8953 - val_loss: 0.9234 - va
Epoch 43/100
32/32 - 0s - 9ms/step - loss: 0.8924 - mse: 0.8924 - val_loss: 0.9201 - val
Epoch 44/100
32/32 - 0s - 10ms/step - loss: 0.8889 - mse: 0.8889 - val_loss: 0.9166 - va
Epoch 45/100
32/32 - 0s - 9ms/step - loss: 0.8861 - mse: 0.8861 - val_loss: 0.9141 - val
Epoch 46/100
32/32 - 0s - 7ms/step - loss: 0.8837 - mse: 0.8837 - val_loss: 0.9113 - val
Epoch 47/100
```

```
32/32 - 0s - 9ms/step - loss: 0.8808 - mse: 0.8808 - val_loss: 0.9095 - val
Epoch 48/100
32/32 - 0s - 10ms/step - loss: 0.8783 - mse: 0.8783 - val_loss: 0.9055 - va
Epoch 49/100
32/32 - 0s - 5ms/step - loss: 0.8757 - mse: 0.8757 - val_loss: 0.9028 - val
Epoch 50/100
32/32 - 0s - 9ms/step - loss: 0.8733 - mse: 0.8733 - val_loss: 0.9009 - val
Epoch 51/100
32/32 - 0s - 10ms/step - loss: 0.8709 - mse: 0.8709 - val_loss: 0.8988 - va
Epoch 52/100
32/32 - 0s - 9ms/step - loss: 0.8687 - mse: 0.8687 - val_loss: 0.8963 - val
Epoch 53/100
32/32 - 0s - 9ms/step - loss: 0.8666 - mse: 0.8666 - val_loss: 0.8939 - val
Epoch 54/100
32/32 - 0s - 9ms/step - loss: 0.8642 - mse: 0.8642 - val_loss: 0.8904 - val
Epoch 55/100
32/32 - 0s - 9ms/step - loss: 0.8620 - mse: 0.8620 - val_loss: 0.8899 - val
Epoch 56/100
32/32 - 0s - 10ms/step - loss: 0.8602 - mse: 0.8602 - val_loss: 0.8863 - va
Epoch 57/100
32/32 - 0s - 5ms/step - loss: 0.8583 - mse: 0.8583 - val_loss: 0.8850 - val
Epoch 58/100
32/32 - 0s - 9ms/step - loss: 0.8562 - mse: 0.8562 - val_loss: 0.8829 - val
Epoch 59/100
32/32 - 0s - 5ms/step - loss: 0.8545 - mse: 0.8545 - val_loss: 0.8804 - val
Epoch 60/100
32/32 - 0s - 10ms/step - loss: 0.8524 - mse: 0.8524 - val_loss: 0.8784 - va
Epoch 61/100
32/32 - 0s - 6ms/step - loss: 0.8508 - mse: 0.8508 - val_loss: 0.8770 - val
Epoch 62/100
32/32 - 0s - 9ms/step - loss: 0.8487 - mse: 0.8487 - val_loss: 0.8734 - val
Epoch 63/100
32/32 - 0s - 9ms/step - loss: 0.8473 - mse: 0.8473 - val_loss: 0.8719 - val
Epoch 64/100
32/32 - 0s - 6ms/step - loss: 0.8454 - mse: 0.8454 - val_loss: 0.8713 - val
Epoch 65/100
32/32 - 0s - 9ms/step - loss: 0.8437 - mse: 0.8437 - val_loss: 0.8697 - val
Epoch 66/100
32/32 - 0s - 5ms/step - loss: 0.8419 - mse: 0.8419 - val_loss: 0.8680 - val
Epoch 67/100
32/32 - 0s - 9ms/step - loss: 0.8402 - mse: 0.8402 - val_loss: 0.8669 - val
Epoch 68/100
32/32 - 0s - 10ms/step - loss: 0.8383 - mse: 0.8383 - val_loss: 0.8640 - va
Epoch 69/100
32/32 - 0s - 9ms/step - loss: 0.8368 - mse: 0.8368 - val_loss: 0.8622 - val
Epoch 70/100
32/32 - 0s - 9ms/step - loss: 0.8350 - mse: 0.8350 - val_loss: 0.8612 - val
Epoch 71/100
32/32 - 0s - 5ms/step - loss: 0.8337 - mse: 0.8337 - val_loss: 0.8586 - val
Epoch 72/100
32/32 - 0s - 9ms/step - loss: 0.8320 - mse: 0.8320 - val_loss: 0.8585 - val
Epoch 73/100
32/32 - 0s - 10ms/step - loss: 0.8306 - mse: 0.8306 - val_loss: 0.8552 - va
Epoch 74/100
```

```
32/32 - 0s - 9ms/step - loss: 0.8288 - mse: 0.8288 - val_loss: 0.8534 - val
Epoch 75/100
32/32 - 0s - 9ms/step - loss: 0.8275 - mse: 0.8275 - val_loss: 0.8517 - val
Epoch 76/100
32/32 - 0s - 9ms/step - loss: 0.8262 - mse: 0.8262 - val_loss: 0.8497 - val
Epoch 77/100
32/32 - 0s - 5ms/step - loss: 0.8244 - mse: 0.8244 - val_loss: 0.8507 - val
Epoch 78/100
32/32 - 0s - 9ms/step - loss: 0.8229 - mse: 0.8229 - val_loss: 0.8471 - val
Epoch 79/100
32/32 - 0s - 5ms/step - loss: 0.8218 - mse: 0.8218 - val_loss: 0.8451 - val
Epoch 80/100
32/32 - 0s - 5ms/step - loss: 0.8203 - mse: 0.8203 - val_loss: 0.8443 - val
Epoch 81/100
32/32 - 0s - 5ms/step - loss: 0.8187 - mse: 0.8187 - val_loss: 0.8415 - val
Epoch 82/100
32/32 - 0s - 5ms/step - loss: 0.8170 - mse: 0.8170 - val_loss: 0.8396 - val
Epoch 83/100
32/32 - 0s - 9ms/step - loss: 0.8158 - mse: 0.8158 - val_loss: 0.8382 - val
Epoch 84/100
32/32 - 0s - 14ms/step - loss: 0.8145 - mse: 0.8145 - val_loss: 0.8374 - va
Epoch 85/100
32/32 - 1s - 16ms/step - loss: 0.8127 - mse: 0.8127 - val_loss: 0.8357 - va
Epoch 86/100
32/32 - 0s - 12ms/step - loss: 0.8113 - mse: 0.8113 - val_loss: 0.8358 - va
Epoch 87/100
32/32 - 1s - 20ms/step - loss: 0.8101 - mse: 0.8101 - val_loss: 0.8338 - va
Epoch 88/100
32/32 - 0s - 15ms/step - loss: 0.8083 - mse: 0.8083 - val_loss: 0.8308 - va
Epoch 89/100
32/32 - 0s - 6ms/step - loss: 0.8074 - mse: 0.8074 - val_loss: 0.8309 - val
Epoch 90/100
32/32 - 0s - 9ms/step - loss: 0.8061 - mse: 0.8061 - val_loss: 0.8299 - val
Epoch 91/100
32/32 - 0s - 5ms/step - loss: 0.8043 - mse: 0.8043 - val_loss: 0.8285 - val
Epoch 92/100
32/32 - 0s - 10ms/step - loss: 0.8028 - mse: 0.8028 - val_loss: 0.8258 - va
Epoch 93/100
32/32 - 0s - 5ms/step - loss: 0.8014 - mse: 0.8014 - val_loss: 0.8244 - val
Epoch 94/100
32/32 - 0s - 5ms/step - loss: 0.8001 - mse: 0.8001 - val_loss: 0.8225 - val
Epoch 95/100
32/32 - 0s - 10ms/step - loss: 0.7988 - mse: 0.7988 - val_loss: 0.8220 - va
Epoch 96/100
32/32 - 0s - 9ms/step - loss: 0.7972 - mse: 0.7972 - val_loss: 0.8186 - val
Epoch 97/100
32/32 - 0s - 9ms/step - loss: 0.7958 - mse: 0.7958 - val_loss: 0.8174 - val
Epoch 98/100
32/32 - 0s - 5ms/step - loss: 0.7940 - mse: 0.7940 - val_loss: 0.8159 - val
Epoch 99/100
32/32 - 0s - 10ms/step - loss: 0.7929 - mse: 0.7929 - val_loss: 0.8155 - va
Epoch 100/100
32/32 - 0s - 5ms/step - loss: 0.7914 - mse: 0.7914 - val_loss: 0.8140 - val
```



```
# Model With Different Activation Functions - 2
model_3 = create_ffnn(activation_functions=['relu', 'relu', 'linear'])
history_3 = train_model(model_3)

# Plot Training & Validation Loss
plt.plot(history_3.history['loss'], label='Train MSE')
plt.plot(history_3.history['val_loss'], label='Validation MSE')
plt.xlabel('Epochs')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

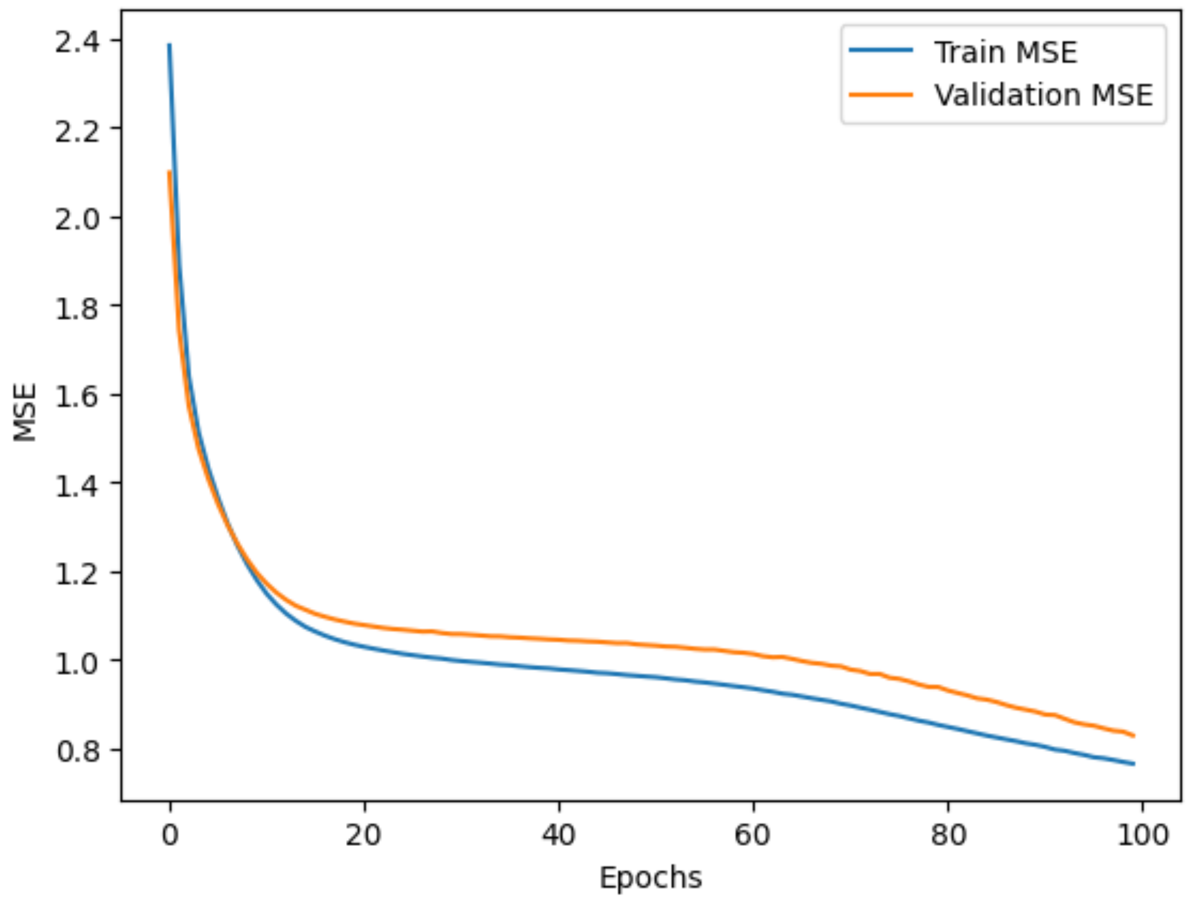
```
Epoch 1/100
32/32 - 1s - 26ms/step - loss: 2.3849 - mse: 2.3849 - val_loss: 2.0982 - va
Epoch 2/100
32/32 - 0s - 13ms/step - loss: 1.8909 - mse: 1.8909 - val_loss: 1.7429 - va
Epoch 3/100
32/32 - 0s - 9ms/step - loss: 1.6396 - mse: 1.6396 - val_loss: 1.5727 - val
Epoch 4/100
32/32 - 0s - 10ms/step - loss: 1.5135 - mse: 1.5135 - val_loss: 1.4764 - va
Epoch 5/100
32/32 - 0s - 9ms/step - loss: 1.4321 - mse: 1.4321 - val_loss: 1.4087 - val
```

```
Epoch 6/100
32/32 - 0s - 5ms/step - loss: 1.3659 - mse: 1.3659 - val_loss: 1.3519 - val
Epoch 7/100
32/32 - 0s - 5ms/step - loss: 1.3074 - mse: 1.3074 - val_loss: 1.3034 - val
Epoch 8/100
32/32 - 0s - 10ms/step - loss: 1.2578 - mse: 1.2578 - val_loss: 1.2623 - va
Epoch 9/100
32/32 - 0s - 10ms/step - loss: 1.2157 - mse: 1.2157 - val_loss: 1.2267 - va
Epoch 10/100
32/32 - 0s - 5ms/step - loss: 1.1800 - mse: 1.1800 - val_loss: 1.1964 - val
Epoch 11/100
32/32 - 0s - 9ms/step - loss: 1.1496 - mse: 1.1496 - val_loss: 1.1723 - val
Epoch 12/100
32/32 - 0s - 10ms/step - loss: 1.1252 - mse: 1.1252 - val_loss: 1.1522 - va
Epoch 13/100
32/32 - 0s - 9ms/step - loss: 1.1049 - mse: 1.1049 - val_loss: 1.1359 - val
Epoch 14/100
32/32 - 0s - 5ms/step - loss: 1.0886 - mse: 1.0886 - val_loss: 1.1225 - val
Epoch 15/100
32/32 - 0s - 5ms/step - loss: 1.0748 - mse: 1.0748 - val_loss: 1.1134 - val
Epoch 16/100
32/32 - 0s - 9ms/step - loss: 1.0644 - mse: 1.0644 - val_loss: 1.1042 - val
Epoch 17/100
32/32 - 0s - 5ms/step - loss: 1.0552 - mse: 1.0552 - val_loss: 1.0977 - val
Epoch 18/100
32/32 - 0s - 5ms/step - loss: 1.0472 - mse: 1.0472 - val_loss: 1.0918 - val
Epoch 19/100
32/32 - 0s - 9ms/step - loss: 1.0405 - mse: 1.0405 - val_loss: 1.0868 - val
Epoch 20/100
32/32 - 0s - 5ms/step - loss: 1.0348 - mse: 1.0348 - val_loss: 1.0823 - val
Epoch 21/100
32/32 - 0s - 10ms/step - loss: 1.0301 - mse: 1.0301 - val_loss: 1.0791 - va
Epoch 22/100
32/32 - 0s - 12ms/step - loss: 1.0257 - mse: 1.0257 - val_loss: 1.0758 - va
Epoch 23/100
32/32 - 0s - 10ms/step - loss: 1.0217 - mse: 1.0217 - val_loss: 1.0726 - va
Epoch 24/100
32/32 - 1s - 16ms/step - loss: 1.0180 - mse: 1.0180 - val_loss: 1.0702 - va
Epoch 25/100
32/32 - 0s - 7ms/step - loss: 1.0142 - mse: 1.0142 - val_loss: 1.0686 - val
Epoch 26/100
32/32 - 0s - 10ms/step - loss: 1.0113 - mse: 1.0113 - val_loss: 1.0664 - va
Epoch 27/100
32/32 - 0s - 9ms/step - loss: 1.0083 - mse: 1.0083 - val_loss: 1.0644 - val
Epoch 28/100
32/32 - 0s - 10ms/step - loss: 1.0057 - mse: 1.0057 - val_loss: 1.0648 - va
Epoch 29/100
32/32 - 0s - 15ms/step - loss: 1.0036 - mse: 1.0036 - val_loss: 1.0612 - va
Epoch 30/100
32/32 - 0s - 6ms/step - loss: 1.0003 - mse: 1.0003 - val_loss: 1.0589 - val
Epoch 31/100
32/32 - 0s - 5ms/step - loss: 0.9981 - mse: 0.9981 - val_loss: 1.0587 - val
Epoch 32/100
32/32 - 0s - 5ms/step - loss: 0.9961 - mse: 0.9961 - val_loss: 1.0572 - val
```

```
Epoch 33/100
32/32 - 0s - 9ms/step - loss: 0.9943 - mse: 0.9943 - val_loss: 1.0556 - val
Epoch 34/100
32/32 - 0s - 9ms/step - loss: 0.9922 - mse: 0.9922 - val_loss: 1.0538 - val
Epoch 35/100
32/32 - 0s - 10ms/step - loss: 0.9900 - mse: 0.9900 - val_loss: 1.0535 - va
Epoch 36/100
32/32 - 0s - 9ms/step - loss: 0.9887 - mse: 0.9887 - val_loss: 1.0517 - val
Epoch 37/100
32/32 - 0s - 9ms/step - loss: 0.9864 - mse: 0.9864 - val_loss: 1.0507 - val
Epoch 38/100
32/32 - 0s - 10ms/step - loss: 0.9844 - mse: 0.9844 - val_loss: 1.0493 - va
Epoch 39/100
32/32 - 0s - 9ms/step - loss: 0.9827 - mse: 0.9827 - val_loss: 1.0482 - val
Epoch 40/100
32/32 - 0s - 9ms/step - loss: 0.9815 - mse: 0.9815 - val_loss: 1.0468 - val
Epoch 41/100
32/32 - 0s - 6ms/step - loss: 0.9792 - mse: 0.9792 - val_loss: 1.0460 - val
Epoch 42/100
32/32 - 0s - 5ms/step - loss: 0.9776 - mse: 0.9776 - val_loss: 1.0442 - val
Epoch 43/100
32/32 - 0s - 6ms/step - loss: 0.9758 - mse: 0.9758 - val_loss: 1.0437 - val
Epoch 44/100
32/32 - 0s - 9ms/step - loss: 0.9739 - mse: 0.9739 - val_loss: 1.0423 - val
Epoch 45/100
32/32 - 0s - 10ms/step - loss: 0.9716 - mse: 0.9716 - val_loss: 1.0416 - va
Epoch 46/100
32/32 - 0s - 9ms/step - loss: 0.9706 - mse: 0.9706 - val_loss: 1.0400 - val
Epoch 47/100
32/32 - 0s - 6ms/step - loss: 0.9686 - mse: 0.9686 - val_loss: 1.0385 - val
Epoch 48/100
32/32 - 0s - 9ms/step - loss: 0.9664 - mse: 0.9664 - val_loss: 1.0385 - val
Epoch 49/100
32/32 - 0s - 10ms/step - loss: 0.9648 - mse: 0.9648 - val_loss: 1.0356 - va
Epoch 50/100
32/32 - 0s - 5ms/step - loss: 0.9631 - mse: 0.9631 - val_loss: 1.0341 - val
Epoch 51/100
32/32 - 0s - 9ms/step - loss: 0.9617 - mse: 0.9617 - val_loss: 1.0328 - val
Epoch 52/100
32/32 - 0s - 9ms/step - loss: 0.9591 - mse: 0.9591 - val_loss: 1.0308 - val
Epoch 53/100
32/32 - 0s - 10ms/step - loss: 0.9565 - mse: 0.9565 - val_loss: 1.0301 - va
Epoch 54/100
32/32 - 0s - 5ms/step - loss: 0.9548 - mse: 0.9548 - val_loss: 1.0280 - val
Epoch 55/100
32/32 - 0s - 5ms/step - loss: 0.9519 - mse: 0.9519 - val_loss: 1.0256 - val
Epoch 56/100
32/32 - 0s - 9ms/step - loss: 0.9499 - mse: 0.9499 - val_loss: 1.0240 - val
Epoch 57/100
32/32 - 0s - 9ms/step - loss: 0.9472 - mse: 0.9472 - val_loss: 1.0240 - val
Epoch 58/100
32/32 - 0s - 10ms/step - loss: 0.9445 - mse: 0.9445 - val_loss: 1.0210 - va
Epoch 59/100
32/32 - 0s - 9ms/step - loss: 0.9416 - mse: 0.9416 - val_loss: 1.0179 - val
```

```
Epoch 60/100
32/32 - 0s - 5ms/step - loss: 0.9390 - mse: 0.9390 - val_loss: 1.0167 - val
Epoch 61/100
32/32 - 0s - 10ms/step - loss: 0.9361 - mse: 0.9361 - val_loss: 1.0139 - va
Epoch 62/100
32/32 - 0s - 5ms/step - loss: 0.9322 - mse: 0.9322 - val_loss: 1.0090 - val
Epoch 63/100
32/32 - 0s - 5ms/step - loss: 0.9288 - mse: 0.9288 - val_loss: 1.0065 - val
Epoch 64/100
32/32 - 0s - 9ms/step - loss: 0.9244 - mse: 0.9244 - val_loss: 1.0072 - val
Epoch 65/100
32/32 - 0s - 5ms/step - loss: 0.9221 - mse: 0.9221 - val_loss: 1.0029 - val
Epoch 66/100
32/32 - 0s - 10ms/step - loss: 0.9182 - mse: 0.9182 - val_loss: 0.9981 - va
Epoch 67/100
32/32 - 0s - 13ms/step - loss: 0.9143 - mse: 0.9143 - val_loss: 0.9940 - va
Epoch 68/100
32/32 - 1s - 16ms/step - loss: 0.9107 - mse: 0.9107 - val_loss: 0.9918 - va
Epoch 69/100
32/32 - 0s - 12ms/step - loss: 0.9071 - mse: 0.9071 - val_loss: 0.9876 - va
Epoch 70/100
32/32 - 1s - 17ms/step - loss: 0.9019 - mse: 0.9019 - val_loss: 0.9860 - va
Epoch 71/100
32/32 - 0s - 5ms/step - loss: 0.8979 - mse: 0.8979 - val_loss: 0.9787 - val
Epoch 72/100
32/32 - 0s - 5ms/step - loss: 0.8930 - mse: 0.8930 - val_loss: 0.9756 - val
Epoch 73/100
32/32 - 0s - 6ms/step - loss: 0.8885 - mse: 0.8885 - val_loss: 0.9689 - val
Epoch 74/100
32/32 - 0s - 6ms/step - loss: 0.8835 - mse: 0.8835 - val_loss: 0.9689 - val
Epoch 75/100
32/32 - 0s - 5ms/step - loss: 0.8784 - mse: 0.8784 - val_loss: 0.9605 - val
Epoch 76/100
32/32 - 0s - 10ms/step - loss: 0.8743 - mse: 0.8743 - val_loss: 0.9578 - va
Epoch 77/100
32/32 - 0s - 9ms/step - loss: 0.8690 - mse: 0.8690 - val_loss: 0.9522 - val
Epoch 78/100
32/32 - 0s - 10ms/step - loss: 0.8638 - mse: 0.8638 - val_loss: 0.9453 - va
Epoch 79/100
32/32 - 0s - 9ms/step - loss: 0.8596 - mse: 0.8596 - val_loss: 0.9399 - val
Epoch 80/100
32/32 - 0s - 5ms/step - loss: 0.8542 - mse: 0.8542 - val_loss: 0.9396 - val
Epoch 81/100
32/32 - 0s - 9ms/step - loss: 0.8495 - mse: 0.8495 - val_loss: 0.9318 - val
Epoch 82/100
32/32 - 0s - 5ms/step - loss: 0.8450 - mse: 0.8450 - val_loss: 0.9260 - val
Epoch 83/100
32/32 - 0s - 9ms/step - loss: 0.8399 - mse: 0.8399 - val_loss: 0.9207 - val
Epoch 84/100
32/32 - 0s - 6ms/step - loss: 0.8351 - mse: 0.8351 - val_loss: 0.9140 - val
Epoch 85/100
32/32 - 0s - 5ms/step - loss: 0.8301 - mse: 0.8301 - val_loss: 0.9111 - val
Epoch 86/100
32/32 - 0s - 9ms/step - loss: 0.8258 - mse: 0.8258 - val_loss: 0.9057 - val
```

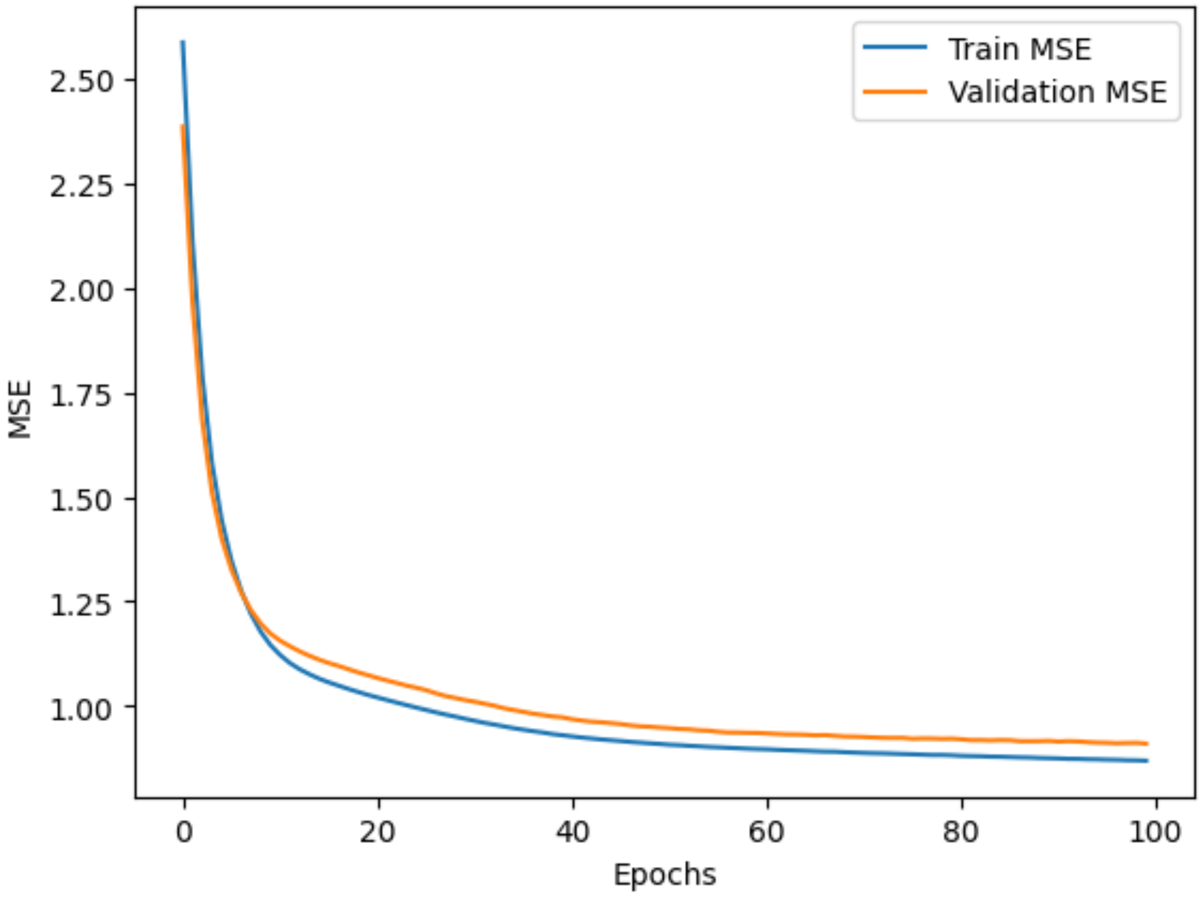
```
Epoch 87/100
32/32 - 0s - 10ms/step - loss: 0.8216 - mse: 0.8216 - val_loss: 0.8988 - va
Epoch 88/100
32/32 - 0s - 5ms/step - loss: 0.8175 - mse: 0.8175 - val_loss: 0.8929 - val
Epoch 89/100
32/32 - 0s - 6ms/step - loss: 0.8127 - mse: 0.8127 - val_loss: 0.8886 - val
Epoch 90/100
32/32 - 0s - 9ms/step - loss: 0.8093 - mse: 0.8093 - val_loss: 0.8841 - val
Epoch 91/100
32/32 - 0s - 10ms/step - loss: 0.8046 - mse: 0.8046 - val_loss: 0.8775 - va
Epoch 92/100
32/32 - 0s - 9ms/step - loss: 0.7985 - mse: 0.7985 - val_loss: 0.8763 - val
Epoch 93/100
32/32 - 0s - 6ms/step - loss: 0.7959 - mse: 0.7959 - val_loss: 0.8679 - val
Epoch 94/100
32/32 - 0s - 9ms/step - loss: 0.7908 - mse: 0.7908 - val_loss: 0.8599 - val
Epoch 95/100
32/32 - 0s - 5ms/step - loss: 0.7866 - mse: 0.7866 - val_loss: 0.8553 - val
Epoch 96/100
32/32 - 0s - 9ms/step - loss: 0.7813 - mse: 0.7813 - val_loss: 0.8523 - val
Epoch 97/100
32/32 - 0s - 9ms/step - loss: 0.7790 - mse: 0.7790 - val_loss: 0.8465 - val
Epoch 98/100
32/32 - 0s - 5ms/step - loss: 0.7750 - mse: 0.7750 - val_loss: 0.8411 - val
Epoch 99/100
32/32 - 0s - 10ms/step - loss: 0.7707 - mse: 0.7707 - val_loss: 0.8389 - va
Epoch 100/100
32/32 - 0s - 9ms/step - loss: 0.7668 - mse: 0.7668 - val_loss: 0.8304 - val
```



```
# Model With Different Activation Functions - 3
model_4 = create_ffnn(activation_functions=['tanh', 'leaky_relu', 'leaky_relu']
history_4 = train_model(model_4)

# Plot Training & Validation Loss
plt.plot(history_4.history['loss'], label='Train MSE')
plt.plot(history_4.history['val_loss'], label='Validation MSE')
plt.xlabel('Epochs')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

```
Epoch 1/100
32/32 - 1s - 26ms/step - loss: 2.5872 - mse: 2.5872 - val_loss: 2.3857 - va
Epoch 2/100
32/32 - 0s - 14ms/step - loss: 2.1113 - mse: 2.1113 - val_loss: 1.9650 - va
Epoch 3/100
32/32 - 0s - 5ms/step - loss: 1.7905 - mse: 1.7905 - val_loss: 1.6835 - val
Epoch 4/100
32/32 - 0s - 5ms/step - loss: 1.5808 - mse: 1.5808 - val_loss: 1.5110 - val
Epoch 5/100
32/32 - 0s - 9ms/step - loss: 1.4450 - mse: 1.4450 - val_loss: 1.4005 - val
Epoch 6/100
32/32 - 0s - 5ms/step - loss: 1.3478 - mse: 1.3478 - val_loss: 1.3262 - val
Epoch 7/100
32/32 - 0s - 10ms/step - loss: 1.2756 - mse: 1.2756 - val_loss: 1.2711 - va
Epoch 8/100
32/32 - 0s - 14ms/step - loss: 1.2205 - mse: 1.2205 - val_loss: 1.2294 - va
Epoch 9/100
32/32 - 1s - 16ms/step - loss: 1.1779 - mse: 1.1779 - val_loss: 1.1973 - va
Epoch 10/100
32/32 - 0s - 9ms/step - loss: 1.1460 - mse: 1.1460 - val_loss: 1.1732 - val
Epoch 11/100
32/32 - 0s - 12ms/step - loss: 1.1219 - mse: 1.1219 - val_loss: 1.1561 - va
Epoch 12/100
32/32 - 0s - 10ms/step - loss: 1.1027 - mse: 1.1027 - val_loss: 1.1428 - va
Epoch 13/100
32/32 - 0s - 15ms/step - loss: 1.0879 - mse: 1.0879 - val_loss: 1.1306 - va
Epoch 14/100
32/32 - 0s - 9ms/step - loss: 1.0759 - mse: 1.0759 - val_loss: 1.1201 - val
Epoch 15/100
32/32 - 0s - 6ms/step - loss: 1.0653 - mse: 1.0653 - val_loss: 1.1107 - val
Epoch 16/100
32/32 - 0s - 5ms/step - loss: 1.0564 - mse: 1.0564 - val_loss: 1.1026 - val
Epoch 17/100
32/32 - 0s - 10ms/step - loss: 1.0485 - mse: 1.0485 - val_loss: 1.0957 - va
Epoch 18/100
```

```
32/32 - 0s - 5ms/step - loss: 1.0407 - mse: 1.0407 - val_loss: 1.0880 - val
Epoch 19/100
32/32 - 0s - 10ms/step - loss: 1.0336 - mse: 1.0336 - val_loss: 1.0804 - va
Epoch 20/100
32/32 - 0s - 10ms/step - loss: 1.0265 - mse: 1.0265 - val_loss: 1.0737 - va
Epoch 21/100
32/32 - 0s - 5ms/step - loss: 1.0202 - mse: 1.0202 - val_loss: 1.0669 - val
Epoch 22/100
32/32 - 0s - 5ms/step - loss: 1.0140 - mse: 1.0140 - val_loss: 1.0607 - val
Epoch 23/100
32/32 - 0s - 10ms/step - loss: 1.0079 - mse: 1.0079 - val_loss: 1.0547 - va
Epoch 24/100
32/32 - 0s - 9ms/step - loss: 1.0019 - mse: 1.0019 - val_loss: 1.0486 - val
Epoch 25/100
32/32 - 0s - 9ms/step - loss: 0.9960 - mse: 0.9960 - val_loss: 1.0435 - val
Epoch 26/100
32/32 - 0s - 9ms/step - loss: 0.9904 - mse: 0.9904 - val_loss: 1.0378 - val
Epoch 27/100
32/32 - 0s - 6ms/step - loss: 0.9847 - mse: 0.9847 - val_loss: 1.0302 - val
Epoch 28/100
32/32 - 0s - 11ms/step - loss: 0.9793 - mse: 0.9793 - val_loss: 1.0237 - va
Epoch 29/100
32/32 - 0s - 8ms/step - loss: 0.9744 - mse: 0.9744 - val_loss: 1.0190 - val
Epoch 30/100
32/32 - 0s - 9ms/step - loss: 0.9691 - mse: 0.9691 - val_loss: 1.0140 - val
Epoch 31/100
32/32 - 0s - 9ms/step - loss: 0.9643 - mse: 0.9643 - val_loss: 1.0100 - val
Epoch 32/100
32/32 - 0s - 6ms/step - loss: 0.9595 - mse: 0.9595 - val_loss: 1.0052 - val
Epoch 33/100
32/32 - 0s - 9ms/step - loss: 0.9554 - mse: 0.9554 - val_loss: 1.0005 - val
Epoch 34/100
32/32 - 0s - 9ms/step - loss: 0.9510 - mse: 0.9510 - val_loss: 0.9944 - val
Epoch 35/100
32/32 - 0s - 9ms/step - loss: 0.9467 - mse: 0.9467 - val_loss: 0.9900 - val
Epoch 36/100
32/32 - 0s - 5ms/step - loss: 0.9430 - mse: 0.9430 - val_loss: 0.9858 - val
Epoch 37/100
32/32 - 0s - 10ms/step - loss: 0.9395 - mse: 0.9395 - val_loss: 0.9814 - va
Epoch 38/100
32/32 - 0s - 9ms/step - loss: 0.9364 - mse: 0.9364 - val_loss: 0.9782 - val
Epoch 39/100
32/32 - 0s - 5ms/step - loss: 0.9325 - mse: 0.9325 - val_loss: 0.9748 - val
Epoch 40/100
32/32 - 0s - 9ms/step - loss: 0.9299 - mse: 0.9299 - val_loss: 0.9727 - val
Epoch 41/100
32/32 - 0s - 5ms/step - loss: 0.9270 - mse: 0.9270 - val_loss: 0.9681 - val
Epoch 42/100
32/32 - 0s - 5ms/step - loss: 0.9244 - mse: 0.9244 - val_loss: 0.9649 - val
Epoch 43/100
32/32 - 0s - 5ms/step - loss: 0.9221 - mse: 0.9221 - val_loss: 0.9622 - val
Epoch 44/100
32/32 - 0s - 5ms/step - loss: 0.9198 - mse: 0.9198 - val_loss: 0.9609 - val
Epoch 45/100
Epoch 45/100
32/32 - 0s - 6ms/step - loss: 0.9179 - mse: 0.9179 - val_loss: 0.9588 - val
Epoch 46/100
32/32 - 0s - 9ms/step - loss: 0.9161 - mse: 0.9161 - val_loss: 0.9566 - val
Epoch 47/100
32/32 - 0s - 9ms/step - loss: 0.9139 - mse: 0.9139 - val_loss: 0.9534 - val
Epoch 48/100
32/32 - 0s - 9ms/step - loss: 0.9122 - mse: 0.9122 - val_loss: 0.9512 - val
Epoch 49/100
32/32 - 0s - 6ms/step - loss: 0.9105 - mse: 0.9105 - val_loss: 0.9502 - val
Epoch 50/100
32/32 - 0s - 5ms/step - loss: 0.9090 - mse: 0.9090 - val_loss: 0.9482 - val
Epoch 51/100
32/32 - 0s - 5ms/step - loss: 0.9071 - mse: 0.9071 - val_loss: 0.9469 - val
Epoch 52/100
32/32 - 0s - 13ms/step - loss: 0.9061 - mse: 0.9061 - val_loss: 0.9449 - va
Epoch 53/100
32/32 - 1s - 16ms/step - loss: 0.9046 - mse: 0.9046 - val_loss: 0.9440 - va
Epoch 54/100
32/32 - 0s - 12ms/step - loss: 0.9034 - mse: 0.9034 - val_loss: 0.9418 - va
Epoch 55/100
32/32 - 1s - 17ms/step - loss: 0.9019 - mse: 0.9019 - val_loss: 0.9408 - va
Epoch 56/100
32/32 - 0s - 9ms/step - loss: 0.9010 - mse: 0.9010 - val_loss: 0.9382 - val
Epoch 57/100
32/32 - 0s - 8ms/step - loss: 0.8998 - mse: 0.8998 - val_loss: 0.9364 - val
Epoch 58/100
32/32 - 0s - 9ms/step - loss: 0.8988 - mse: 0.8988 - val_loss: 0.9363 - val
Epoch 59/100
32/32 - 0s - 5ms/step - loss: 0.8977 - mse: 0.8977 - val_loss: 0.9356 - val
Epoch 60/100
32/32 - 0s - 10ms/step - loss: 0.8968 - mse: 0.8968 - val_loss: 0.9353 - va
Epoch 61/100
32/32 - 0s - 6ms/step - loss: 0.8962 - mse: 0.8962 - val_loss: 0.9342 - val
Epoch 62/100
32/32 - 0s - 5ms/step - loss: 0.8951 - mse: 0.8951 - val_loss: 0.9330 - val
Epoch 63/100
32/32 - 0s - 9ms/step - loss: 0.8943 - mse: 0.8943 - val_loss: 0.9319 - val
Epoch 64/100
32/32 - 0s - 6ms/step - loss: 0.8934 - mse: 0.8934 - val_loss: 0.9316 - val
Epoch 65/100
32/32 - 0s - 5ms/step - loss: 0.8926 - mse: 0.8926 - val_loss: 0.9310 - val
Epoch 66/100
32/32 - 0s - 5ms/step - loss: 0.8916 - mse: 0.8916 - val_loss: 0.9295 - val
Epoch 67/100
32/32 - 0s - 6ms/step - loss: 0.8907 - mse: 0.8907 - val_loss: 0.9302 - val
Epoch 68/100
32/32 - 0s - 9ms/step - loss: 0.8905 - mse: 0.8905 - val_loss: 0.9284 - val
Epoch 69/100
32/32 - 0s - 9ms/step - loss: 0.8892 - mse: 0.8892 - val_loss: 0.9268 - val
Epoch 70/100
32/32 - 0s - 5ms/step - loss: 0.8883 - mse: 0.8883 - val_loss: 0.9265 - val
Epoch 71/100
32/32 - 0s - 6ms/step - loss: 0.8875 - mse: 0.8875 - val_loss: 0.9256 - val
Epoch 72/100
Epoch 72/100
32/32 - 0s - 5ms/step - loss: 0.8869 - mse: 0.8869 - val_loss: 0.9245 - val
Epoch 73/100
32/32 - 0s - 9ms/step - loss: 0.8865 - mse: 0.8865 - val_loss: 0.9237 - val
Epoch 74/100
32/32 - 0s - 10ms/step - loss: 0.8857 - mse: 0.8857 - val_loss: 0.9233 - va
Epoch 75/100
32/32 - 0s - 5ms/step - loss: 0.8849 - mse: 0.8849 - val_loss: 0.9236 - val
Epoch 76/100
32/32 - 0s - 5ms/step - loss: 0.8843 - mse: 0.8843 - val_loss: 0.9213 - val
Epoch 77/100
32/32 - 0s - 5ms/step - loss: 0.8834 - mse: 0.8834 - val_loss: 0.9221 - val
Epoch 78/100
32/32 - 0s - 9ms/step - loss: 0.8826 - mse: 0.8826 - val_loss: 0.9220 - val
Epoch 79/100
32/32 - 0s - 10ms/step - loss: 0.8826 - mse: 0.8826 - val_loss: 0.9211 - va
Epoch 80/100
32/32 - 0s - 9ms/step - loss: 0.8819 - mse: 0.8819 - val_loss: 0.9216 - val
Epoch 81/100
32/32 - 0s - 9ms/step - loss: 0.8806 - mse: 0.8806 - val_loss: 0.9202 - val
Epoch 82/100
32/32 - 0s - 6ms/step - loss: 0.8800 - mse: 0.8800 - val_loss: 0.9181 - val
Epoch 83/100
32/32 - 0s - 9ms/step - loss: 0.8796 - mse: 0.8796 - val_loss: 0.9183 - val
Epoch 84/100
32/32 - 0s - 5ms/step - loss: 0.8787 - mse: 0.8787 - val_loss: 0.9175 - val
Epoch 85/100
32/32 - 0s - 9ms/step - loss: 0.8783 - mse: 0.8783 - val_loss: 0.9182 - val
Epoch 86/100
32/32 - 0s - 6ms/step - loss: 0.8776 - mse: 0.8776 - val_loss: 0.9179 - val
Epoch 87/100
32/32 - 0s - 5ms/step - loss: 0.8769 - mse: 0.8769 - val_loss: 0.9157 - val
Epoch 88/100
32/32 - 0s - 9ms/step - loss: 0.8765 - mse: 0.8765 - val_loss: 0.9155 - val
Epoch 89/100
32/32 - 0s - 9ms/step - loss: 0.8755 - mse: 0.8755 - val_loss: 0.9156 - val
Epoch 90/100
32/32 - 0s - 5ms/step - loss: 0.8750 - mse: 0.8750 - val_loss: 0.9163 - val
Epoch 91/100
32/32 - 0s - 5ms/step - loss: 0.8743 - mse: 0.8743 - val_loss: 0.9146 - val
Epoch 92/100
32/32 - 0s - 6ms/step - loss: 0.8734 - mse: 0.8734 - val_loss: 0.9156 - val
Epoch 93/100
32/32 - 0s - 9ms/step - loss: 0.8732 - mse: 0.8732 - val_loss: 0.9150 - val
Epoch 94/100
32/32 - 0s - 9ms/step - loss: 0.8724 - mse: 0.8724 - val_loss: 0.9133 - val
Epoch 95/100
32/32 - 0s - 9ms/step - loss: 0.8721 - mse: 0.8721 - val_loss: 0.9122 - val
Epoch 96/100
32/32 - 0s - 9ms/step - loss: 0.8713 - mse: 0.8713 - val_loss: 0.9117 - val
Epoch 97/100
32/32 - 0s - 9ms/step - loss: 0.8710 - mse: 0.8710 - val_loss: 0.9107 - val
Epoch 98/100
32/32 - 0s - 11ms/step - loss: 0.8701 - mse: 0.8701 - val_loss: 0.9112 - va
Epoch 99/100
Epoch 99/100
32/32 - 0s - 10ms/step - loss: 0.8697 - mse: 0.8697 - val_loss: 0.9116 - va
Epoch 100/100
32/32 - 1s - 16ms/step - loss: 0.8689 - mse: 0.8689 - val_loss: 0.9098 - va
```



```
# Model with Different Learning Rate - 1
model_5 = create_ffnn()
history_5 = train_model(model_5,learning_rate=0.05)

# Plot Training & Validation Loss
plt.plot(history_5.history['loss'], label='Train MSE')
plt.plot(history_5.history['val_loss'], label='Validation MSE')
plt.xlabel('Epochs')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

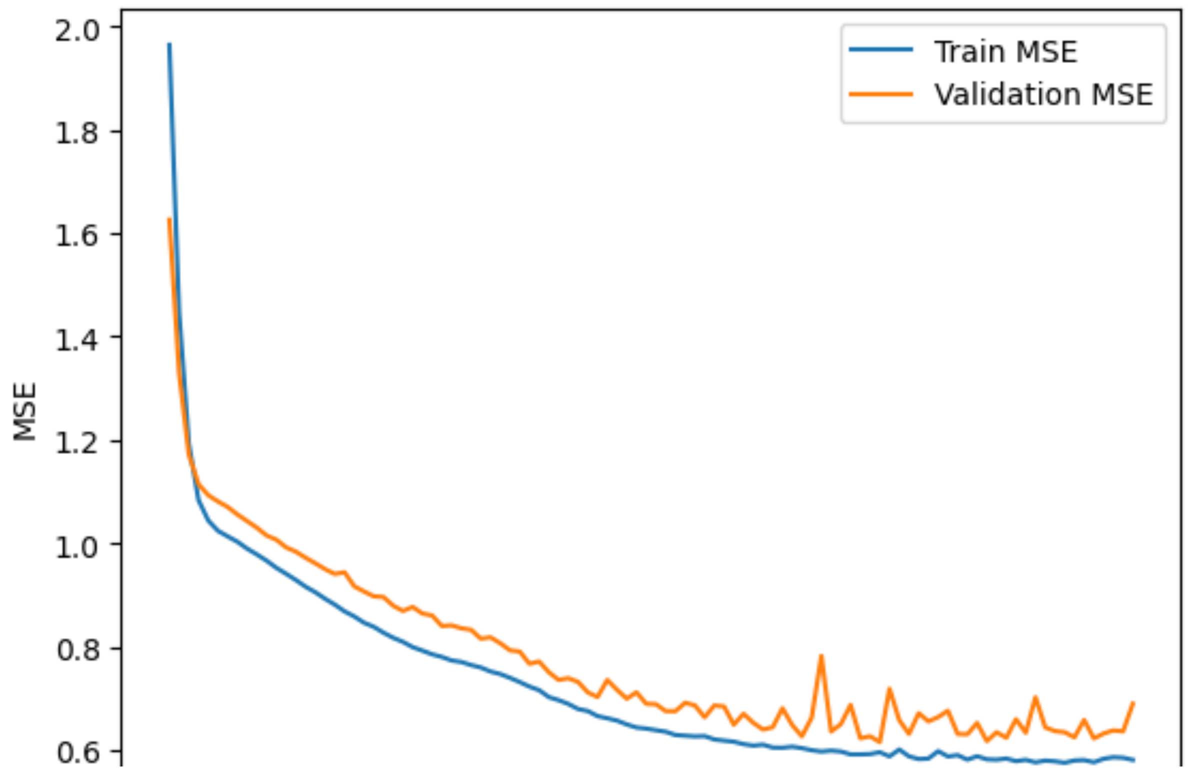
```
Epoch 1/100
32/32 - 1s - 33ms/step - loss: 1.9638 - mse: 1.9638 - val_loss: 1.6256 - va
Epoch 2/100
32/32 - 0s - 6ms/step - loss: 1.4469 - mse: 1.4469 - val_loss: 1.3296 - val
Epoch 3/100
32/32 - 0s - 6ms/step - loss: 1.1932 - mse: 1.1932 - val_loss: 1.1717 - val
```

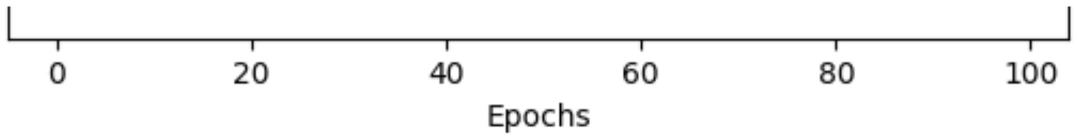
```
32/32 - 0s - 6ms/step - loss: 1.1932 - mse: 1.1932 - val_loss: 1.1717 - val
Epoch 4/100
32/32 - 0s - 9ms/step - loss: 1.0835 - mse: 1.0835 - val_loss: 1.1141 - val
Epoch 5/100
32/32 - 0s - 9ms/step - loss: 1.0441 - mse: 1.0441 - val_loss: 1.0930 - val
Epoch 6/100
32/32 - 0s - 5ms/step - loss: 1.0241 - mse: 1.0241 - val_loss: 1.0809 - val
Epoch 7/100
32/32 - 0s - 9ms/step - loss: 1.0133 - mse: 1.0133 - val_loss: 1.0702 - val
Epoch 8/100
32/32 - 0s - 5ms/step - loss: 1.0030 - mse: 1.0030 - val_loss: 1.0557 - val
Epoch 9/100
32/32 - 0s - 5ms/step - loss: 0.9896 - mse: 0.9896 - val_loss: 1.0429 - val
Epoch 10/100
32/32 - 0s - 6ms/step - loss: 0.9784 - mse: 0.9784 - val_loss: 1.0301 - val
Epoch 11/100
32/32 - 0s - 5ms/step - loss: 0.9664 - mse: 0.9664 - val_loss: 1.0151 - val
Epoch 12/100
32/32 - 0s - 10ms/step - loss: 0.9526 - mse: 0.9526 - val_loss: 1.0069 - va
Epoch 13/100
32/32 - 0s - 9ms/step - loss: 0.9411 - mse: 0.9411 - val_loss: 0.9921 - val
Epoch 14/100
32/32 - 0s - 6ms/step - loss: 0.9291 - mse: 0.9291 - val_loss: 0.9839 - val
Epoch 15/100
32/32 - 0s - 9ms/step - loss: 0.9161 - mse: 0.9161 - val_loss: 0.9721 - val
Epoch 16/100
32/32 - 0s - 5ms/step - loss: 0.9054 - mse: 0.9054 - val_loss: 0.9614 - val
Epoch 17/100
32/32 - 0s - 9ms/step - loss: 0.8928 - mse: 0.8928 - val_loss: 0.9501 - val
Epoch 18/100
32/32 - 0s - 5ms/step - loss: 0.8815 - mse: 0.8815 - val_loss: 0.9407 - val
Epoch 19/100
32/32 - 0s - 5ms/step - loss: 0.8685 - mse: 0.8685 - val_loss: 0.9441 - val
Epoch 20/100
32/32 - 0s - 10ms/step - loss: 0.8586 - mse: 0.8586 - val_loss: 0.9169 - va
Epoch 21/100
32/32 - 0s - 9ms/step - loss: 0.8465 - mse: 0.8465 - val_loss: 0.9069 - val
Epoch 22/100
32/32 - 0s - 9ms/step - loss: 0.8386 - mse: 0.8386 - val_loss: 0.8976 - val
Epoch 23/100
32/32 - 0s - 9ms/step - loss: 0.8272 - mse: 0.8272 - val_loss: 0.8962 - val
Epoch 24/100
32/32 - 0s - 10ms/step - loss: 0.8174 - mse: 0.8174 - val_loss: 0.8794 - va
Epoch 25/100
32/32 - 0s - 9ms/step - loss: 0.8095 - mse: 0.8095 - val_loss: 0.8693 - val
Epoch 26/100
32/32 - 0s - 5ms/step - loss: 0.7993 - mse: 0.7993 - val_loss: 0.8770 - val
Epoch 27/100
32/32 - 0s - 9ms/step - loss: 0.7926 - mse: 0.7926 - val_loss: 0.8643 - val
Epoch 28/100
32/32 - 0s - 6ms/step - loss: 0.7855 - mse: 0.7855 - val_loss: 0.8605 - val
Epoch 29/100
32/32 - 0s - 9ms/step - loss: 0.7804 - mse: 0.7804 - val_loss: 0.8402 - val
Epoch 30/100
32/32 - 0s - 9ms/step - loss: 0.7735 - mse: 0.7735 - val_loss: 0.8414 - val
```

```
32/32 - 0s - 9ms/step - loss: 0.7755 - mse: 0.7755 - val_loss: 0.8414 - val
Epoch 31/100
32/32 - 0s - 9ms/step - loss: 0.7706 - mse: 0.7706 - val_loss: 0.8357 - val
Epoch 32/100
32/32 - 0s - 10ms/step - loss: 0.7647 - mse: 0.7647 - val_loss: 0.8329 - va
Epoch 33/100
32/32 - 0s - 9ms/step - loss: 0.7600 - mse: 0.7600 - val_loss: 0.8156 - val
Epoch 34/100
32/32 - 0s - 9ms/step - loss: 0.7522 - mse: 0.7522 - val_loss: 0.8183 - val
Epoch 35/100
32/32 - 0s - 9ms/step - loss: 0.7473 - mse: 0.7473 - val_loss: 0.8069 - val
Epoch 36/100
32/32 - 0s - 10ms/step - loss: 0.7399 - mse: 0.7399 - val_loss: 0.7932 - va
Epoch 37/100
32/32 - 0s - 5ms/step - loss: 0.7318 - mse: 0.7318 - val_loss: 0.7907 - val
Epoch 38/100
32/32 - 0s - 10ms/step - loss: 0.7230 - mse: 0.7230 - val_loss: 0.7672 - va
Epoch 39/100
32/32 - 0s - 6ms/step - loss: 0.7161 - mse: 0.7161 - val_loss: 0.7715 - val
Epoch 40/100
32/32 - 0s - 13ms/step - loss: 0.7026 - mse: 0.7026 - val_loss: 0.7507 - va
Epoch 41/100
32/32 - 1s - 17ms/step - loss: 0.6968 - mse: 0.6968 - val_loss: 0.7359 - va
Epoch 42/100
32/32 - 0s - 10ms/step - loss: 0.6895 - mse: 0.6895 - val_loss: 0.7392 - va
Epoch 43/100
32/32 - 0s - 9ms/step - loss: 0.6797 - mse: 0.6797 - val_loss: 0.7320 - val
Epoch 44/100
32/32 - 0s - 10ms/step - loss: 0.6757 - mse: 0.6757 - val_loss: 0.7117 - va
Epoch 45/100
32/32 - 0s - 8ms/step - loss: 0.6660 - mse: 0.6660 - val_loss: 0.7029 - val
Epoch 46/100
32/32 - 0s - 8ms/step - loss: 0.6618 - mse: 0.6618 - val_loss: 0.7359 - val
Epoch 47/100
32/32 - 0s - 5ms/step - loss: 0.6573 - mse: 0.6573 - val_loss: 0.7167 - val
Epoch 48/100
32/32 - 0s - 10ms/step - loss: 0.6498 - mse: 0.6498 - val_loss: 0.6997 - va
Epoch 49/100
32/32 - 0s - 9ms/step - loss: 0.6439 - mse: 0.6439 - val_loss: 0.7119 - val
Epoch 50/100
32/32 - 0s - 9ms/step - loss: 0.6418 - mse: 0.6418 - val_loss: 0.6898 - val
Epoch 51/100
32/32 - 0s - 10ms/step - loss: 0.6387 - mse: 0.6387 - val_loss: 0.6889 - va
Epoch 52/100
32/32 - 0s - 9ms/step - loss: 0.6358 - mse: 0.6358 - val_loss: 0.6756 - val
Epoch 53/100
32/32 - 0s - 10ms/step - loss: 0.6294 - mse: 0.6294 - val_loss: 0.6754 - va
Epoch 54/100
32/32 - 0s - 9ms/step - loss: 0.6281 - mse: 0.6281 - val_loss: 0.6917 - val
Epoch 55/100
32/32 - 0s - 10ms/step - loss: 0.6267 - mse: 0.6267 - val_loss: 0.6867 - va
Epoch 56/100
32/32 - 0s - 5ms/step - loss: 0.6273 - mse: 0.6273 - val_loss: 0.6637 - val
Epoch 57/100
32/32 - 0s - 5ms/step - loss: 0.6208 - mse: 0.6208 - val_loss: 0.6865 - val
```

```
32/32 - 0s - 5ms/step - loss: 0.6208 - mse: 0.6208 - val_loss: 0.6865 - val
Epoch 58/100
32/32 - 0s - 6ms/step - loss: 0.6186 - mse: 0.6186 - val_loss: 0.6845 - val
Epoch 59/100
32/32 - 0s - 5ms/step - loss: 0.6165 - mse: 0.6165 - val_loss: 0.6494 - val
Epoch 60/100
32/32 - 0s - 9ms/step - loss: 0.6118 - mse: 0.6118 - val_loss: 0.6704 - val
Epoch 61/100
32/32 - 0s - 10ms/step - loss: 0.6089 - mse: 0.6089 - val_loss: 0.6530 - va
Epoch 62/100
32/32 - 0s - 9ms/step - loss: 0.6107 - mse: 0.6107 - val_loss: 0.6400 - val
Epoch 63/100
32/32 - 0s - 6ms/step - loss: 0.6051 - mse: 0.6051 - val_loss: 0.6444 - val
Epoch 64/100
32/32 - 0s - 9ms/step - loss: 0.6048 - mse: 0.6048 - val_loss: 0.6811 - val
Epoch 65/100
32/32 - 0s - 9ms/step - loss: 0.6068 - mse: 0.6068 - val_loss: 0.6487 - val
Epoch 66/100
32/32 - 0s - 10ms/step - loss: 0.6039 - mse: 0.6039 - val_loss: 0.6269 - va
Epoch 67/100
32/32 - 0s - 5ms/step - loss: 0.6000 - mse: 0.6000 - val_loss: 0.6629 - val
Epoch 68/100
32/32 - 0s - 9ms/step - loss: 0.5975 - mse: 0.5975 - val_loss: 0.7823 - val
Epoch 69/100
32/32 - 0s - 6ms/step - loss: 0.5992 - mse: 0.5992 - val_loss: 0.6362 - val
Epoch 70/100
32/32 - 0s - 10ms/step - loss: 0.5975 - mse: 0.5975 - val_loss: 0.6503 - va
Epoch 71/100
32/32 - 0s - 10ms/step - loss: 0.5920 - mse: 0.5920 - val_loss: 0.6876 - va
Epoch 72/100
32/32 - 0s - 8ms/step - loss: 0.5919 - mse: 0.5919 - val_loss: 0.6231 - val
Epoch 73/100
32/32 - 0s - 10ms/step - loss: 0.5925 - mse: 0.5925 - val_loss: 0.6269 - va
Epoch 74/100
32/32 - 0s - 9ms/step - loss: 0.5961 - mse: 0.5961 - val_loss: 0.6157 - val
Epoch 75/100
32/32 - 0s - 6ms/step - loss: 0.5880 - mse: 0.5880 - val_loss: 0.7188 - val
Epoch 76/100
32/32 - 0s - 6ms/step - loss: 0.6014 - mse: 0.6014 - val_loss: 0.6582 - val
Epoch 77/100
32/32 - 0s - 9ms/step - loss: 0.5881 - mse: 0.5881 - val_loss: 0.6315 - val
Epoch 78/100
32/32 - 0s - 9ms/step - loss: 0.5830 - mse: 0.5830 - val_loss: 0.6715 - val
Epoch 79/100
32/32 - 0s - 5ms/step - loss: 0.5839 - mse: 0.5839 - val_loss: 0.6558 - val
Epoch 80/100
32/32 - 0s - 5ms/step - loss: 0.5982 - mse: 0.5982 - val_loss: 0.6640 - val
Epoch 81/100
32/32 - 0s - 6ms/step - loss: 0.5879 - mse: 0.5879 - val_loss: 0.6763 - val
Epoch 82/100
32/32 - 0s - 6ms/step - loss: 0.5905 - mse: 0.5905 - val_loss: 0.6315 - val
Epoch 83/100
32/32 - 0s - 9ms/step - loss: 0.5817 - mse: 0.5817 - val_loss: 0.6308 - val
Epoch 84/100
32/32 - 0s - 15ms/step - loss: 0.5884 - mse: 0.5884 - val_loss: 0.6537 - va
```

```
32/32 - 0s - 15ms/step - loss: 0.5884 - mse: 0.5884 - val_loss: 0.6527 - va
Epoch 85/100
32/32 - 0s - 15ms/step - loss: 0.5824 - mse: 0.5824 - val_loss: 0.6176 - va
Epoch 86/100
32/32 - 0s - 10ms/step - loss: 0.5816 - mse: 0.5816 - val_loss: 0.6347 - va
Epoch 87/100
32/32 - 0s - 11ms/step - loss: 0.5839 - mse: 0.5839 - val_loss: 0.6240 - va
Epoch 88/100
32/32 - 0s - 10ms/step - loss: 0.5788 - mse: 0.5788 - val_loss: 0.6599 - va
Epoch 89/100
32/32 - 0s - 15ms/step - loss: 0.5814 - mse: 0.5814 - val_loss: 0.6338 - va
Epoch 90/100
32/32 - 0s - 5ms/step - loss: 0.5765 - mse: 0.5765 - val_loss: 0.7024 - val
Epoch 91/100
32/32 - 0s - 5ms/step - loss: 0.5800 - mse: 0.5800 - val_loss: 0.6441 - val
Epoch 92/100
32/32 - 0s - 9ms/step - loss: 0.5785 - mse: 0.5785 - val_loss: 0.6370 - val
Epoch 93/100
32/32 - 0s - 10ms/step - loss: 0.5757 - mse: 0.5757 - val_loss: 0.6348 - va
Epoch 94/100
32/32 - 0s - 8ms/step - loss: 0.5801 - mse: 0.5801 - val_loss: 0.6249 - val
Epoch 95/100
32/32 - 0s - 5ms/step - loss: 0.5809 - mse: 0.5809 - val_loss: 0.6584 - val
Epoch 96/100
32/32 - 0s - 9ms/step - loss: 0.5767 - mse: 0.5767 - val_loss: 0.6228 - val
Epoch 97/100
32/32 - 0s - 6ms/step - loss: 0.5834 - mse: 0.5834 - val_loss: 0.6322 - val
Epoch 98/100
32/32 - 0s - 6ms/step - loss: 0.5865 - mse: 0.5865 - val_loss: 0.6383 - val
Epoch 99/100
32/32 - 0s - 5ms/step - loss: 0.5854 - mse: 0.5854 - val_loss: 0.6365 - val
Epoch 100/100
32/32 - 0s - 5ms/step - loss: 0.5813 - mse: 0.5813 - val_loss: 0.6898 - val
```



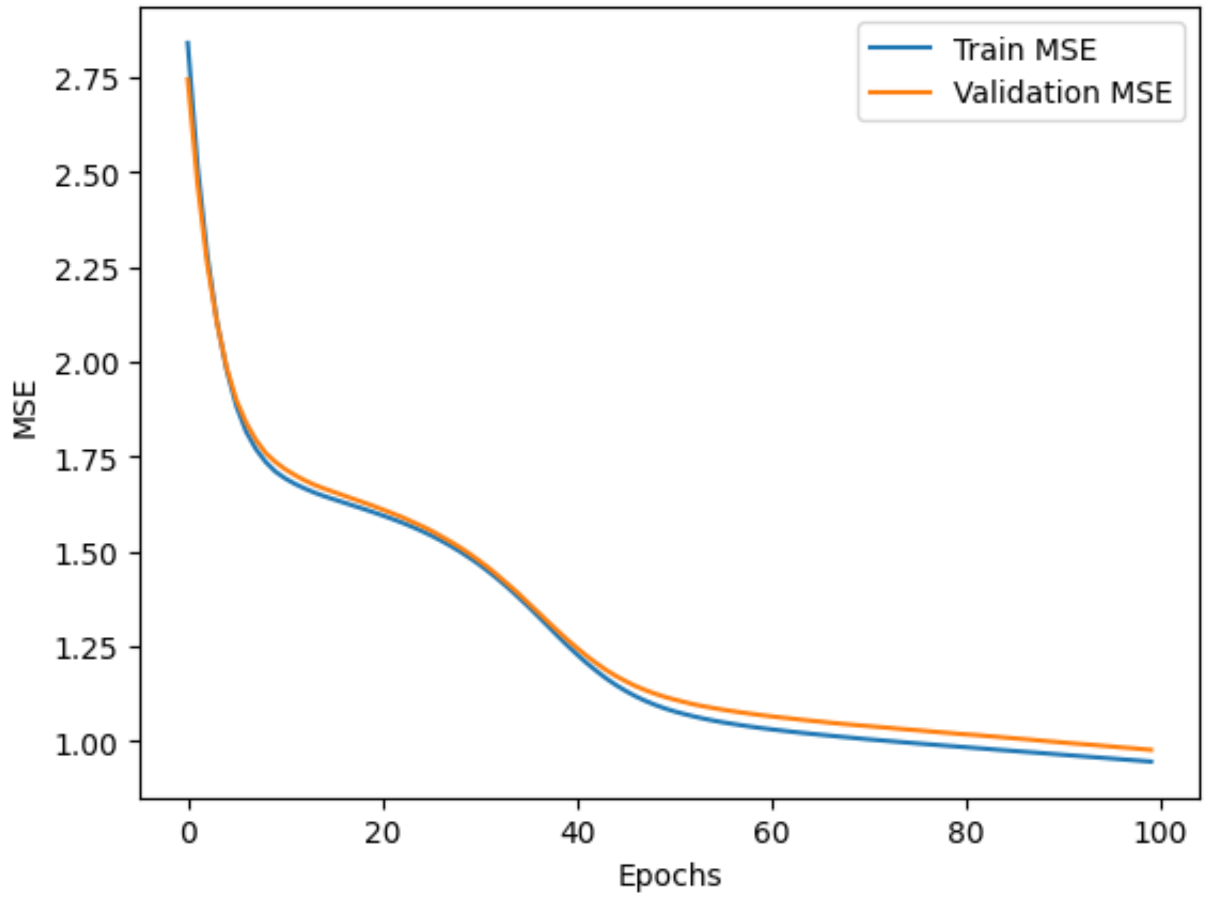


```
# Model with Different Learning Rate - 2
model_6 = create_ffnn()
history_6 = train_model(model_6,learning_rate=0.005)

# Plot Training & Validation Loss
plt.plot(history_6.history['loss'], label='Train MSE')
plt.plot(history_6.history['val_loss'], label='Validation MSE')
plt.xlabel('Epochs')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

```
Epoch 1/100
32/32 - 1s - 25ms/step - loss: 2.8400 - mse: 2.8400 - val_loss: 2.7441 - va
Epoch 2/100
32/32 - 0s - 14ms/step - loss: 2.5176 - mse: 2.5176 - val_loss: 2.4689 - va
Epoch 3/100
32/32 - 0s - 5ms/step - loss: 2.2821 - mse: 2.2821 - val_loss: 2.2586 - val
Epoch 4/100
32/32 - 0s - 5ms/step - loss: 2.1065 - mse: 2.1065 - val_loss: 2.1033 - val
Epoch 5/100
32/32 - 0s - 9ms/step - loss: 1.9784 - mse: 1.9784 - val_loss: 1.9843 - val
Epoch 6/100
32/32 - 0s - 10ms/step - loss: 1.8831 - mse: 1.8831 - val_loss: 1.8985 - va
Epoch 7/100
32/32 - 0s - 10ms/step - loss: 1.8162 - mse: 1.8162 - val_loss: 1.8395 - va
Epoch 8/100
32/32 - 0s - 9ms/step - loss: 1.7704 - mse: 1.7704 - val_loss: 1.7944 - val
Epoch 9/100
32/32 - 0s - 9ms/step - loss: 1.7364 - mse: 1.7364 - val_loss: 1.7599 - val
Epoch 10/100
32/32 - 0s - 6ms/step - loss: 1.7109 - mse: 1.7109 - val_loss: 1.7363 - val
Epoch 11/100
32/32 - 0s - 6ms/step - loss: 1.6930 - mse: 1.6930 - val_loss: 1.7173 - val
Epoch 12/100
32/32 - 0s - 5ms/step - loss: 1.6786 - mse: 1.6786 - val_loss: 1.7017 - val
Epoch 13/100
32/32 - 0s - 9ms/step - loss: 1.6664 - mse: 1.6664 - val_loss: 1.6880 - val
Epoch 14/100
32/32 - 0s - 9ms/step - loss: 1.6556 - mse: 1.6556 - val_loss: 1.6761 - val
Epoch 15/100
32/32 - 0s - 9ms/step - loss: 1.6462 - mse: 1.6462 - val_loss: 1.6661 - val
Epoch 16/100
Epoch 16/100
32/32 - 0s - 5ms/step - loss: 1.6378 - mse: 1.6378 - val_loss: 1.6571 - val
Epoch 17/100
32/32 - 0s - 5ms/step - loss: 1.6296 - mse: 1.6296 - val_loss: 1.6480 - val
Epoch 18/100
32/32 - 0s - 5ms/step - loss: 1.6212 - mse: 1.6212 - val_loss: 1.6389 - val
Epoch 19/100
32/32 - 0s - 10ms/step - loss: 1.6130 - mse: 1.6130 - val_loss: 1.6300 - va
Epoch 20/100
32/32 - 0s - 9ms/step - loss: 1.6044 - mse: 1.6044 - val_loss: 1.6209 - val
Epoch 21/100
32/32 - 0s - 10ms/step - loss: 1.5955 - mse: 1.5955 - val_loss: 1.6111 - va
Epoch 22/100
32/32 - 0s - 9ms/step - loss: 1.5863 - mse: 1.5863 - val_loss: 1.6011 - val
Epoch 23/100
32/32 - 0s - 10ms/step - loss: 1.5766 - mse: 1.5766 - val_loss: 1.5910 - va
Epoch 24/100
32/32 - 0s - 9ms/step - loss: 1.5662 - mse: 1.5662 - val_loss: 1.5793 - val
Epoch 25/100
32/32 - 0s - 5ms/step - loss: 1.5548 - mse: 1.5548 - val_loss: 1.5678 - val
Epoch 26/100
32/32 - 0s - 11ms/step - loss: 1.5427 - mse: 1.5427 - val_loss: 1.5553 - va
Epoch 27/100
32/32 - 0s - 7ms/step - loss: 1.5295 - mse: 1.5295 - val_loss: 1.5411 - val
Epoch 28/100
32/32 - 0s - 9ms/step - loss: 1.5152 - mse: 1.5152 - val_loss: 1.5264 - val
Epoch 29/100
32/32 - 0s - 9ms/step - loss: 1.4998 - mse: 1.4998 - val_loss: 1.5108 - val
Epoch 30/100
32/32 - 0s - 10ms/step - loss: 1.4831 - mse: 1.4831 - val_loss: 1.4938 - va
Epoch 31/100
32/32 - 0s - 11ms/step - loss: 1.4653 - mse: 1.4653 - val_loss: 1.4751 - va
Epoch 32/100
32/32 - 0s - 15ms/step - loss: 1.4456 - mse: 1.4456 - val_loss: 1.4557 - va
Epoch 33/100
32/32 - 0s - 10ms/step - loss: 1.4250 - mse: 1.4250 - val_loss: 1.4343 - va
Epoch 34/100
32/32 - 0s - 5ms/step - loss: 1.4028 - mse: 1.4028 - val_loss: 1.4120 - val
Epoch 35/100
32/32 - 0s - 9ms/step - loss: 1.3795 - mse: 1.3795 - val_loss: 1.3893 - val
Epoch 36/100
32/32 - 0s - 10ms/step - loss: 1.3552 - mse: 1.3552 - val_loss: 1.3647 - va
Epoch 37/100
32/32 - 0s - 5ms/step - loss: 1.3298 - mse: 1.3298 - val_loss: 1.3405 - val
Epoch 38/100
32/32 - 0s - 5ms/step - loss: 1.3042 - mse: 1.3042 - val_loss: 1.3157 - val
Epoch 39/100
32/32 - 0s - 5ms/step - loss: 1.2783 - mse: 1.2783 - val_loss: 1.2916 - val
Epoch 40/100
32/32 - 0s - 6ms/step - loss: 1.2532 - mse: 1.2532 - val_loss: 1.2682 - val
Epoch 41/100
32/32 - 0s - 9ms/step - loss: 1.2292 - mse: 1.2292 - val_loss: 1.2461 - val
Epoch 42/100
32/32 - 0s - 10ms/step - loss: 1.2065 - mse: 1.2065 - val_loss: 1.2250 - va
Epoch 43/100
Epoch 43/100
32/32 - 0s - 5ms/step - loss: 1.1856 - mse: 1.1856 - val_loss: 1.2058 - val
Epoch 44/100
32/32 - 0s - 9ms/step - loss: 1.1660 - mse: 1.1660 - val_loss: 1.1881 - val
Epoch 45/100
32/32 - 0s - 9ms/step - loss: 1.1486 - mse: 1.1486 - val_loss: 1.1725 - val
Epoch 46/100
32/32 - 0s - 10ms/step - loss: 1.1332 - mse: 1.1332 - val_loss: 1.1586 - va
Epoch 47/100
32/32 - 0s - 9ms/step - loss: 1.1192 - mse: 1.1192 - val_loss: 1.1460 - val
Epoch 48/100
32/32 - 0s - 9ms/step - loss: 1.1072 - mse: 1.1072 - val_loss: 1.1354 - val
Epoch 49/100
32/32 - 0s - 5ms/step - loss: 1.0965 - mse: 1.0965 - val_loss: 1.1258 - val
Epoch 50/100
32/32 - 0s - 9ms/step - loss: 1.0872 - mse: 1.0872 - val_loss: 1.1175 - val
Epoch 51/100
32/32 - 0s - 9ms/step - loss: 1.0791 - mse: 1.0791 - val_loss: 1.1104 - val
Epoch 52/100
32/32 - 0s - 5ms/step - loss: 1.0723 - mse: 1.0723 - val_loss: 1.1038 - val
Epoch 53/100
32/32 - 0s - 9ms/step - loss: 1.0659 - mse: 1.0659 - val_loss: 1.0977 - val
Epoch 54/100
32/32 - 0s - 5ms/step - loss: 1.0601 - mse: 1.0601 - val_loss: 1.0926 - val
Epoch 55/100
32/32 - 0s - 5ms/step - loss: 1.0550 - mse: 1.0550 - val_loss: 1.0879 - val
Epoch 56/100
32/32 - 0s - 6ms/step - loss: 1.0503 - mse: 1.0503 - val_loss: 1.0835 - val
Epoch 57/100
32/32 - 0s - 9ms/step - loss: 1.0461 - mse: 1.0461 - val_loss: 1.0796 - val
Epoch 58/100
32/32 - 0s - 5ms/step - loss: 1.0422 - mse: 1.0422 - val_loss: 1.0758 - val
Epoch 59/100
32/32 - 0s - 5ms/step - loss: 1.0384 - mse: 1.0384 - val_loss: 1.0724 - val
Epoch 60/100
32/32 - 0s - 9ms/step - loss: 1.0349 - mse: 1.0349 - val_loss: 1.0689 - val
Epoch 61/100
32/32 - 0s - 5ms/step - loss: 1.0317 - mse: 1.0317 - val_loss: 1.0659 - val
Epoch 62/100
32/32 - 0s - 10ms/step - loss: 1.0287 - mse: 1.0287 - val_loss: 1.0629 - va
Epoch 63/100
32/32 - 0s - 10ms/step - loss: 1.0257 - mse: 1.0257 - val_loss: 1.0603 - va
Epoch 64/100
32/32 - 0s - 6ms/step - loss: 1.0229 - mse: 1.0229 - val_loss: 1.0572 - val
Epoch 65/100
32/32 - 0s - 6ms/step - loss: 1.0202 - mse: 1.0202 - val_loss: 1.0546 - val
Epoch 66/100
32/32 - 0s - 9ms/step - loss: 1.0176 - mse: 1.0176 - val_loss: 1.0522 - val
Epoch 67/100
32/32 - 0s - 10ms/step - loss: 1.0152 - mse: 1.0152 - val_loss: 1.0493 - va
Epoch 68/100
32/32 - 0s - 10ms/step - loss: 1.0126 - mse: 1.0126 - val_loss: 1.0469 - va
Epoch 69/100
32/32 - 0s - 5ms/step - loss: 1.0103 - mse: 1.0103 - val_loss: 1.0446 - val
```

```
Epoch 70/100
32/32 - 0s - 5ms/step - loss: 1.0078 - mse: 1.0078 - val_loss: 1.0423 - val
Epoch 71/100
32/32 - 0s - 9ms/step - loss: 1.0054 - mse: 1.0054 - val_loss: 1.0402 - val
Epoch 72/100
32/32 - 0s - 11ms/step - loss: 1.0033 - mse: 1.0033 - val_loss: 1.0380 - va
Epoch 73/100
32/32 - 0s - 9ms/step - loss: 1.0012 - mse: 1.0012 - val_loss: 1.0359 - val
Epoch 74/100
32/32 - 0s - 10ms/step - loss: 0.9990 - mse: 0.9990 - val_loss: 1.0334 - va
Epoch 75/100
32/32 - 0s - 9ms/step - loss: 0.9968 - mse: 0.9968 - val_loss: 1.0313 - val
Epoch 76/100
32/32 - 0s - 9ms/step - loss: 0.9947 - mse: 0.9947 - val_loss: 1.0294 - val
Epoch 77/100
32/32 - 0s - 11ms/step - loss: 0.9927 - mse: 0.9927 - val_loss: 1.0270 - va
Epoch 78/100
32/32 - 0s - 12ms/step - loss: 0.9906 - mse: 0.9906 - val_loss: 1.0248 - va
Epoch 79/100
32/32 - 0s - 15ms/step - loss: 0.9885 - mse: 0.9885 - val_loss: 1.0227 - va
Epoch 80/100
32/32 - 0s - 9ms/step - loss: 0.9866 - mse: 0.9866 - val_loss: 1.0205 - val
Epoch 81/100
32/32 - 0s - 5ms/step - loss: 0.9846 - mse: 0.9846 - val_loss: 1.0185 - val
Epoch 82/100
32/32 - 0s - 9ms/step - loss: 0.9826 - mse: 0.9826 - val_loss: 1.0168 - val
Epoch 83/100
32/32 - 0s - 10ms/step - loss: 0.9807 - mse: 0.9807 - val_loss: 1.0145 - va
Epoch 84/100
32/32 - 0s - 6ms/step - loss: 0.9786 - mse: 0.9786 - val_loss: 1.0125 - val
Epoch 85/100
32/32 - 0s - 9ms/step - loss: 0.9766 - mse: 0.9766 - val_loss: 1.0102 - val
Epoch 86/100
32/32 - 0s - 5ms/step - loss: 0.9746 - mse: 0.9746 - val_loss: 1.0081 - val
Epoch 87/100
32/32 - 0s - 10ms/step - loss: 0.9727 - mse: 0.9727 - val_loss: 1.0059 - va
Epoch 88/100
32/32 - 0s - 9ms/step - loss: 0.9708 - mse: 0.9708 - val_loss: 1.0037 - val
Epoch 89/100
32/32 - 0s - 6ms/step - loss: 0.9687 - mse: 0.9687 - val_loss: 1.0012 - val
Epoch 90/100
32/32 - 0s - 9ms/step - loss: 0.9668 - mse: 0.9668 - val_loss: 0.9992 - val
Epoch 91/100
32/32 - 0s - 5ms/step - loss: 0.9647 - mse: 0.9647 - val_loss: 0.9968 - val
Epoch 92/100
32/32 - 0s - 5ms/step - loss: 0.9627 - mse: 0.9627 - val_loss: 0.9947 - val
Epoch 93/100
32/32 - 0s - 6ms/step - loss: 0.9608 - mse: 0.9608 - val_loss: 0.9927 - val
Epoch 94/100
32/32 - 0s - 9ms/step - loss: 0.9587 - mse: 0.9587 - val_loss: 0.9908 - val
Epoch 95/100
32/32 - 0s - 6ms/step - loss: 0.9569 - mse: 0.9569 - val_loss: 0.9885 - val
Epoch 96/100
32/32 - 0s - 6ms/step - loss: 0.9548 - mse: 0.9548 - val_loss: 0.9864 - val
```

```
Epoch 97/100
32/32 - 0s - 9ms/step - loss: 0.9528 - mse: 0.9528 - val_loss: 0.9841 - val
Epoch 98/100
32/32 - 0s - 6ms/step - loss: 0.9508 - mse: 0.9508 - val_loss: 0.9822 - val
Epoch 99/100
32/32 - 0s - 9ms/step - loss: 0.9488 - mse: 0.9488 - val_loss: 0.9799 - val
Epoch 100/100
32/32 - 0s - 9ms/step - loss: 0.9468 - mse: 0.9468 - val_loss: 0.9777 - val
```



```
# Model with Different Learning Rate - 3
model_7 = create_ffnn()
history_7 = train_model(model_7,learning_rate=0.001)

# Plot Training & Validation Loss
plt.plot(history_7.history['loss'], label='Train MSE')
plt.plot(history_7.history['val_loss'], label='Validation MSE')
plt.xlabel('Epochs')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

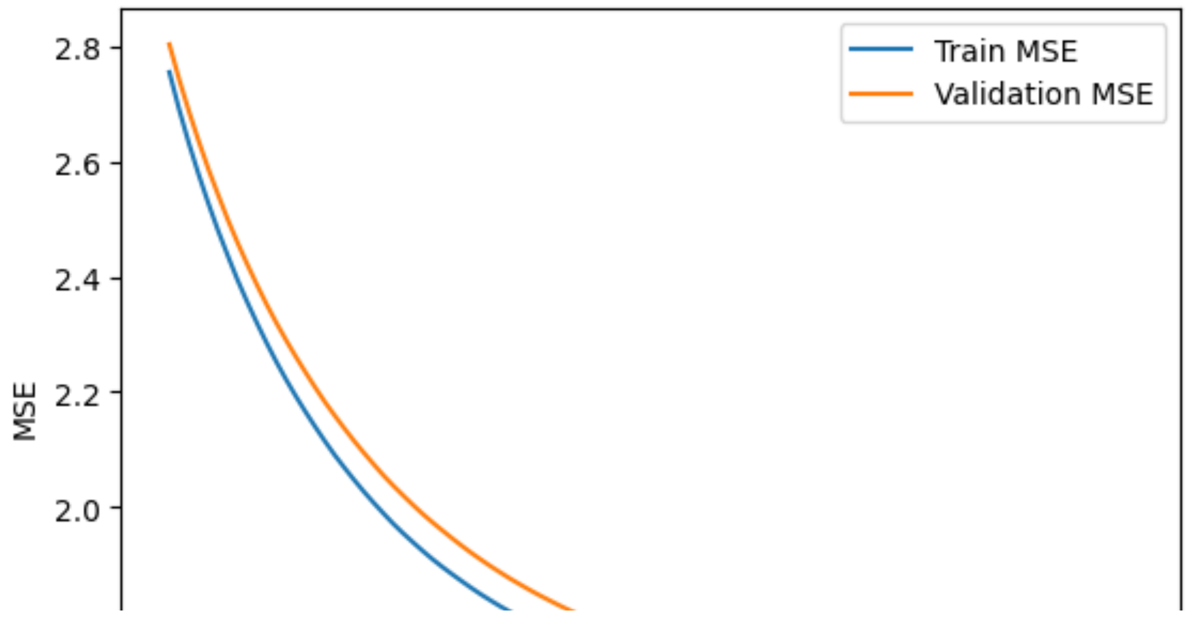
```
Epoch 1/100
```

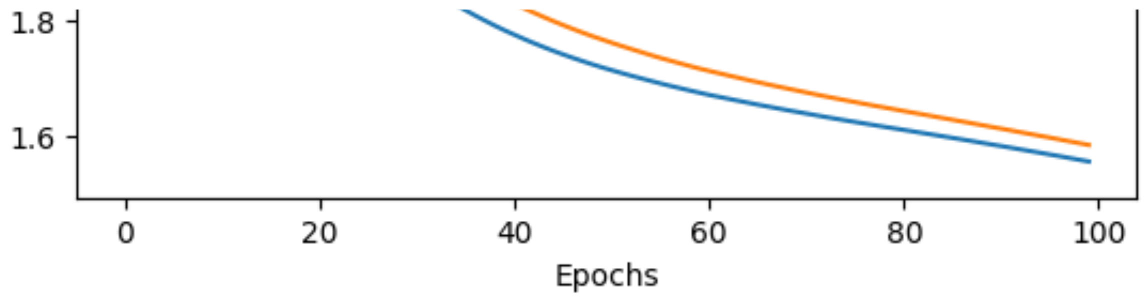
```
32/32 - 1s - 25ms/step - loss: 2.7551 - mse: 2.7551 - val_loss: 2.8034 - va
Epoch 2/100
32/32 - 0s - 6ms/step - loss: 2.6908 - mse: 2.6908 - val_loss: 2.7430 - val
Epoch 3/100
32/32 - 0s - 9ms/step - loss: 2.6319 - mse: 2.6319 - val_loss: 2.6876 - val
Epoch 4/100
32/32 - 0s - 10ms/step - loss: 2.5779 - mse: 2.5779 - val_loss: 2.6360 - va
Epoch 5/100
32/32 - 0s - 10ms/step - loss: 2.5276 - mse: 2.5276 - val_loss: 2.5873 - va
Epoch 6/100
32/32 - 0s - 9ms/step - loss: 2.4802 - mse: 2.4802 - val_loss: 2.5425 - val
Epoch 7/100
32/32 - 0s - 9ms/step - loss: 2.4366 - mse: 2.4366 - val_loss: 2.4991 - val
Epoch 8/100
32/32 - 0s - 5ms/step - loss: 2.3944 - mse: 2.3944 - val_loss: 2.4592 - val
Epoch 9/100
32/32 - 0s - 5ms/step - loss: 2.3558 - mse: 2.3558 - val_loss: 2.4215 - val
Epoch 10/100
32/32 - 0s - 9ms/step - loss: 2.3193 - mse: 2.3193 - val_loss: 2.3856 - val
Epoch 11/100
32/32 - 0s - 10ms/step - loss: 2.2845 - mse: 2.2845 - val_loss: 2.3515 - va
Epoch 12/100
32/32 - 0s - 9ms/step - loss: 2.2517 - mse: 2.2517 - val_loss: 2.3195 - val
Epoch 13/100
32/32 - 0s - 9ms/step - loss: 2.2210 - mse: 2.2210 - val_loss: 2.2894 - val
Epoch 14/100
32/32 - 0s - 8ms/step - loss: 2.1921 - mse: 2.1921 - val_loss: 2.2608 - val
Epoch 15/100
32/32 - 0s - 11ms/step - loss: 2.1648 - mse: 2.1648 - val_loss: 2.2334 - va
Epoch 16/100
32/32 - 0s - 7ms/step - loss: 2.1387 - mse: 2.1387 - val_loss: 2.2070 - val
Epoch 17/100
32/32 - 0s - 12ms/step - loss: 2.1136 - mse: 2.1136 - val_loss: 2.1818 - va
Epoch 18/100
32/32 - 1s - 17ms/step - loss: 2.0897 - mse: 2.0897 - val_loss: 2.1584 - va
Epoch 19/100
32/32 - 0s - 10ms/step - loss: 2.0678 - mse: 2.0678 - val_loss: 2.1359 - va
Epoch 20/100
32/32 - 0s - 15ms/step - loss: 2.0465 - mse: 2.0465 - val_loss: 2.1141 - va
Epoch 21/100
32/32 - 0s - 9ms/step - loss: 2.0261 - mse: 2.0261 - val_loss: 2.0936 - val
Epoch 22/100
32/32 - 0s - 10ms/step - loss: 2.0070 - mse: 2.0070 - val_loss: 2.0738 - va
Epoch 23/100
32/32 - 0s - 5ms/step - loss: 1.9886 - mse: 1.9886 - val_loss: 2.0546 - val
Epoch 24/100
32/32 - 0s - 5ms/step - loss: 1.9708 - mse: 1.9708 - val_loss: 2.0369 - val
Epoch 25/100
32/32 - 0s - 9ms/step - loss: 1.9544 - mse: 1.9544 - val_loss: 2.0201 - val
Epoch 26/100
32/32 - 0s - 6ms/step - loss: 1.9389 - mse: 1.9389 - val_loss: 2.0037 - val
Epoch 27/100
32/32 - 0s - 5ms/step - loss: 1.9239 - mse: 1.9239 - val_loss: 1.9881 - val
Epoch 28/100
```

```
32/32 - 0s - 9ms/step - loss: 1.9096 - mse: 1.9096 - val_loss: 1.9735 - val
Epoch 29/100
32/32 - 0s - 5ms/step - loss: 1.8963 - mse: 1.8963 - val_loss: 1.9597 - val
Epoch 30/100
32/32 - 0s - 6ms/step - loss: 1.8836 - mse: 1.8836 - val_loss: 1.9461 - val
Epoch 31/100
32/32 - 0s - 6ms/step - loss: 1.8714 - mse: 1.8714 - val_loss: 1.9329 - val
Epoch 32/100
32/32 - 0s - 5ms/step - loss: 1.8594 - mse: 1.8594 - val_loss: 1.9203 - val
Epoch 33/100
32/32 - 0s - 5ms/step - loss: 1.8481 - mse: 1.8481 - val_loss: 1.9084 - val
Epoch 34/100
32/32 - 0s - 9ms/step - loss: 1.8375 - mse: 1.8375 - val_loss: 1.8969 - val
Epoch 35/100
32/32 - 0s - 9ms/step - loss: 1.8272 - mse: 1.8272 - val_loss: 1.8861 - val
Epoch 36/100
32/32 - 0s - 5ms/step - loss: 1.8175 - mse: 1.8175 - val_loss: 1.8755 - val
Epoch 37/100
32/32 - 0s - 9ms/step - loss: 1.8081 - mse: 1.8081 - val_loss: 1.8655 - val
Epoch 38/100
32/32 - 0s - 9ms/step - loss: 1.7993 - mse: 1.7993 - val_loss: 1.8558 - val
Epoch 39/100
32/32 - 0s - 10ms/step - loss: 1.7908 - mse: 1.7908 - val_loss: 1.8466 - va
Epoch 40/100
32/32 - 0s - 9ms/step - loss: 1.7827 - mse: 1.7827 - val_loss: 1.8379 - val
Epoch 41/100
32/32 - 0s - 9ms/step - loss: 1.7751 - mse: 1.7751 - val_loss: 1.8293 - val
Epoch 42/100
32/32 - 0s - 9ms/step - loss: 1.7676 - mse: 1.7676 - val_loss: 1.8212 - val
Epoch 43/100
32/32 - 0s - 9ms/step - loss: 1.7605 - mse: 1.7605 - val_loss: 1.8133 - val
Epoch 44/100
32/32 - 0s - 10ms/step - loss: 1.7537 - mse: 1.7537 - val_loss: 1.8058 - va
Epoch 45/100
32/32 - 0s - 9ms/step - loss: 1.7471 - mse: 1.7471 - val_loss: 1.7987 - val
Epoch 46/100
32/32 - 0s - 10ms/step - loss: 1.7409 - mse: 1.7409 - val_loss: 1.7918 - va
Epoch 47/100
32/32 - 0s - 5ms/step - loss: 1.7350 - mse: 1.7350 - val_loss: 1.7852 - val
Epoch 48/100
32/32 - 0s - 10ms/step - loss: 1.7293 - mse: 1.7293 - val_loss: 1.7788 - va
Epoch 49/100
32/32 - 0s - 9ms/step - loss: 1.7238 - mse: 1.7238 - val_loss: 1.7726 - val
Epoch 50/100
32/32 - 0s - 5ms/step - loss: 1.7185 - mse: 1.7185 - val_loss: 1.7669 - val
Epoch 51/100
32/32 - 0s - 9ms/step - loss: 1.7136 - mse: 1.7136 - val_loss: 1.7611 - val
Epoch 52/100
32/32 - 0s - 5ms/step - loss: 1.7087 - mse: 1.7087 - val_loss: 1.7556 - val
Epoch 53/100
32/32 - 0s - 10ms/step - loss: 1.7040 - mse: 1.7040 - val_loss: 1.7503 - va
Epoch 54/100
32/32 - 0s - 9ms/step - loss: 1.6995 - mse: 1.6995 - val_loss: 1.7451 - val
Epoch 55/100
```

```
32/32 - 0s - 10ms/step - loss: 1.6951 - mse: 1.6951 - val_loss: 1.7398 - va
Epoch 56/100
32/32 - 0s - 12ms/step - loss: 1.6907 - mse: 1.6907 - val_loss: 1.7348 - va
Epoch 57/100
32/32 - 0s - 7ms/step - loss: 1.6864 - mse: 1.6864 - val_loss: 1.7302 - val
Epoch 58/100
32/32 - 0s - 10ms/step - loss: 1.6825 - mse: 1.6825 - val_loss: 1.7255 - va
Epoch 59/100
32/32 - 0s - 7ms/step - loss: 1.6785 - mse: 1.6785 - val_loss: 1.7209 - val
Epoch 60/100
32/32 - 0s - 9ms/step - loss: 1.6746 - mse: 1.6746 - val_loss: 1.7165 - val
Epoch 61/100
32/32 - 0s - 10ms/step - loss: 1.6708 - mse: 1.6708 - val_loss: 1.7124 - va
Epoch 62/100
32/32 - 0s - 11ms/step - loss: 1.6673 - mse: 1.6673 - val_loss: 1.7084 - va
Epoch 63/100
32/32 - 1s - 20ms/step - loss: 1.6639 - mse: 1.6639 - val_loss: 1.7044 - va
Epoch 64/100
32/32 - 0s - 15ms/step - loss: 1.6605 - mse: 1.6605 - val_loss: 1.7005 - va
Epoch 65/100
32/32 - 0s - 5ms/step - loss: 1.6572 - mse: 1.6572 - val_loss: 1.6966 - val
Epoch 66/100
32/32 - 0s - 10ms/step - loss: 1.6539 - mse: 1.6539 - val_loss: 1.6928 - va
Epoch 67/100
32/32 - 0s - 5ms/step - loss: 1.6507 - mse: 1.6507 - val_loss: 1.6891 - val
Epoch 68/100
32/32 - 0s - 9ms/step - loss: 1.6475 - mse: 1.6475 - val_loss: 1.6855 - val
Epoch 69/100
32/32 - 0s - 10ms/step - loss: 1.6444 - mse: 1.6444 - val_loss: 1.6820 - va
Epoch 70/100
32/32 - 0s - 10ms/step - loss: 1.6414 - mse: 1.6414 - val_loss: 1.6785 - va
Epoch 71/100
32/32 - 0s - 5ms/step - loss: 1.6384 - mse: 1.6384 - val_loss: 1.6749 - val
Epoch 72/100
32/32 - 0s - 9ms/step - loss: 1.6354 - mse: 1.6354 - val_loss: 1.6714 - val
Epoch 73/100
32/32 - 0s - 10ms/step - loss: 1.6323 - mse: 1.6323 - val_loss: 1.6680 - va
Epoch 74/100
32/32 - 0s - 9ms/step - loss: 1.6294 - mse: 1.6294 - val_loss: 1.6647 - val
Epoch 75/100
32/32 - 0s - 5ms/step - loss: 1.6266 - mse: 1.6266 - val_loss: 1.6615 - val
Epoch 76/100
32/32 - 0s - 9ms/step - loss: 1.6238 - mse: 1.6238 - val_loss: 1.6583 - val
Epoch 77/100
32/32 - 0s - 9ms/step - loss: 1.6210 - mse: 1.6210 - val_loss: 1.6551 - val
Epoch 78/100
32/32 - 0s - 9ms/step - loss: 1.6182 - mse: 1.6182 - val_loss: 1.6519 - val
Epoch 79/100
32/32 - 0s - 10ms/step - loss: 1.6154 - mse: 1.6154 - val_loss: 1.6489 - va
Epoch 80/100
32/32 - 0s - 9ms/step - loss: 1.6127 - mse: 1.6127 - val_loss: 1.6459 - val
Epoch 81/100
32/32 - 0s - 9ms/step - loss: 1.6100 - mse: 1.6100 - val_loss: 1.6428 - val
Epoch 82/100
```

```
32/32 - 0s - 9ms/step - loss: 1.6072 - mse: 1.6072 - val_loss: 1.6397 - val
Epoch 83/100
32/32 - 0s - 9ms/step - loss: 1.6045 - mse: 1.6045 - val_loss: 1.6367 - val
Epoch 84/100
32/32 - 0s - 10ms/step - loss: 1.6017 - mse: 1.6017 - val_loss: 1.6337 - va
Epoch 85/100
32/32 - 0s - 6ms/step - loss: 1.5990 - mse: 1.5990 - val_loss: 1.6306 - val
Epoch 86/100
32/32 - 0s - 9ms/step - loss: 1.5962 - mse: 1.5962 - val_loss: 1.6276 - val
Epoch 87/100
32/32 - 0s - 6ms/step - loss: 1.5934 - mse: 1.5934 - val_loss: 1.6245 - val
Epoch 88/100
32/32 - 0s - 9ms/step - loss: 1.5906 - mse: 1.5906 - val_loss: 1.6214 - val
Epoch 89/100
32/32 - 0s - 6ms/step - loss: 1.5877 - mse: 1.5877 - val_loss: 1.6183 - val
Epoch 90/100
32/32 - 0s - 9ms/step - loss: 1.5848 - mse: 1.5848 - val_loss: 1.6152 - val
Epoch 91/100
32/32 - 0s - 9ms/step - loss: 1.5820 - mse: 1.5820 - val_loss: 1.6122 - val
Epoch 92/100
32/32 - 0s - 9ms/step - loss: 1.5791 - mse: 1.5791 - val_loss: 1.6091 - val
Epoch 93/100
32/32 - 0s - 6ms/step - loss: 1.5762 - mse: 1.5762 - val_loss: 1.6059 - val
Epoch 94/100
32/32 - 0s - 9ms/step - loss: 1.5733 - mse: 1.5733 - val_loss: 1.6028 - val
Epoch 95/100
32/32 - 0s - 9ms/step - loss: 1.5703 - mse: 1.5703 - val_loss: 1.5997 - val
Epoch 96/100
32/32 - 0s - 5ms/step - loss: 1.5674 - mse: 1.5674 - val_loss: 1.5967 - val
Epoch 97/100
32/32 - 0s - 6ms/step - loss: 1.5644 - mse: 1.5644 - val_loss: 1.5935 - val
Epoch 98/100
32/32 - 0s - 9ms/step - loss: 1.5613 - mse: 1.5613 - val_loss: 1.5903 - val
Epoch 99/100
32/32 - 0s - 9ms/step - loss: 1.5582 - mse: 1.5582 - val_loss: 1.5871 - val
Epoch 100/100
32/32 - 0s - 12ms/step - loss: 1.5552 - mse: 1.5552 - val_loss: 1.5839 - va
```





```
# Model with Different Epochs - 1
model_8 = create_ffnn(activation_functions=['tanh', 'leaky_relu', 'linear'])
history_8 = train_model(model_8,learning_rate=0.01,epochs=150)

# Plot Training & Validation Loss
plt.plot(history_8.history['loss'], label='Train MSE')
plt.plot(history_8.history['val_loss'], label='Validation MSE')
plt.xlabel('Epochs')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

```
Epoch 1/150
32/32 - 1s - 25ms/step - loss: 2.3490 - mse: 2.3490 - val_loss: 2.1559 - va
Epoch 2/150
32/32 - 0s - 14ms/step - loss: 1.9825 - mse: 1.9825 - val_loss: 1.8590 - va
Epoch 3/150
32/32 - 0s - 9ms/step - loss: 1.7399 - mse: 1.7399 - val_loss: 1.6373 - val
Epoch 4/150
32/32 - 0s - 9ms/step - loss: 1.5616 - mse: 1.5616 - val_loss: 1.4779 - val
Epoch 5/150
32/32 - 0s - 10ms/step - loss: 1.4296 - mse: 1.4296 - val_loss: 1.3669 - va
Epoch 6/150
32/32 - 0s - 5ms/step - loss: 1.3331 - mse: 1.3331 - val_loss: 1.2824 - val
Epoch 7/150
32/32 - 0s - 10ms/step - loss: 1.2555 - mse: 1.2555 - val_loss: 1.2143 - va
Epoch 8/150
32/32 - 0s - 5ms/step - loss: 1.1905 - mse: 1.1905 - val_loss: 1.1644 - val
Epoch 9/150
32/32 - 0s - 5ms/step - loss: 1.1395 - mse: 1.1395 - val_loss: 1.1252 - val
Epoch 10/150
32/32 - 0s - 10ms/step - loss: 1.0995 - mse: 1.0995 - val_loss: 1.0958 - va
Epoch 11/150
32/32 - 0s - 9ms/step - loss: 1.0683 - mse: 1.0683 - val_loss: 1.0747 - val
Epoch 12/150
32/32 - 0s - 9ms/step - loss: 1.0446 - mse: 1.0446 - val_loss: 1.0566 - val
Epoch 13/150
32/32 - 0s - 9ms/step - loss: 1.0253 - mse: 1.0253 - val_loss: 1.0427 - val
```

```
Epoch 14/150
32/32 - 0s - 10ms/step - loss: 1.0094 - mse: 1.0094 - val_loss: 1.0303 - va
Epoch 15/150
32/32 - 0s - 9ms/step - loss: 0.9966 - mse: 0.9966 - val_loss: 1.0214 - val
Epoch 16/150
32/32 - 0s - 10ms/step - loss: 0.9859 - mse: 0.9859 - val_loss: 1.0132 - va
Epoch 17/150
32/32 - 0s - 9ms/step - loss: 0.9765 - mse: 0.9765 - val_loss: 1.0052 - val
Epoch 18/150
32/32 - 0s - 9ms/step - loss: 0.9683 - mse: 0.9683 - val_loss: 0.9991 - val
Epoch 19/150
32/32 - 0s - 5ms/step - loss: 0.9607 - mse: 0.9607 - val_loss: 0.9934 - val
Epoch 20/150
32/32 - 0s - 6ms/step - loss: 0.9541 - mse: 0.9541 - val_loss: 0.9883 - val
Epoch 21/150
32/32 - 0s - 13ms/step - loss: 0.9476 - mse: 0.9476 - val_loss: 0.9827 - va
Epoch 22/150
32/32 - 1s - 17ms/step - loss: 0.9423 - mse: 0.9423 - val_loss: 0.9778 - va
Epoch 23/150
32/32 - 0s - 12ms/step - loss: 0.9369 - mse: 0.9369 - val_loss: 0.9736 - va
Epoch 24/150
32/32 - 1s - 17ms/step - loss: 0.9318 - mse: 0.9318 - val_loss: 0.9686 - va
Epoch 25/150
32/32 - 0s - 7ms/step - loss: 0.9273 - mse: 0.9273 - val_loss: 0.9633 - val
Epoch 26/150
32/32 - 0s - 5ms/step - loss: 0.9226 - mse: 0.9226 - val_loss: 0.9600 - val
Epoch 27/150
32/32 - 0s - 5ms/step - loss: 0.9187 - mse: 0.9187 - val_loss: 0.9553 - val
Epoch 28/150
32/32 - 0s - 5ms/step - loss: 0.9146 - mse: 0.9146 - val_loss: 0.9505 - val
Epoch 29/150
32/32 - 0s - 6ms/step - loss: 0.9106 - mse: 0.9106 - val_loss: 0.9467 - val
Epoch 30/150
32/32 - 0s - 5ms/step - loss: 0.9067 - mse: 0.9067 - val_loss: 0.9428 - val
Epoch 31/150
32/32 - 0s - 5ms/step - loss: 0.9028 - mse: 0.9028 - val_loss: 0.9395 - val
Epoch 32/150
32/32 - 0s - 10ms/step - loss: 0.8990 - mse: 0.8990 - val_loss: 0.9354 - va
Epoch 33/150
32/32 - 0s - 10ms/step - loss: 0.8962 - mse: 0.8962 - val_loss: 0.9323 - va
Epoch 34/150
32/32 - 0s - 8ms/step - loss: 0.8926 - mse: 0.8926 - val_loss: 0.9296 - val
Epoch 35/150
32/32 - 0s - 10ms/step - loss: 0.8895 - mse: 0.8895 - val_loss: 0.9262 - va
Epoch 36/150
32/32 - 0s - 9ms/step - loss: 0.8867 - mse: 0.8867 - val_loss: 0.9217 - val
Epoch 37/150
32/32 - 0s - 9ms/step - loss: 0.8844 - mse: 0.8844 - val_loss: 0.9194 - val
Epoch 38/150
32/32 - 0s - 5ms/step - loss: 0.8819 - mse: 0.8819 - val_loss: 0.9171 - val
Epoch 39/150
32/32 - 0s - 6ms/step - loss: 0.8791 - mse: 0.8791 - val_loss: 0.9145 - val
Epoch 40/150
32/32 - 0s - 9ms/step - loss: 0.8772 - mse: 0.8772 - val_loss: 0.9125 - val
```

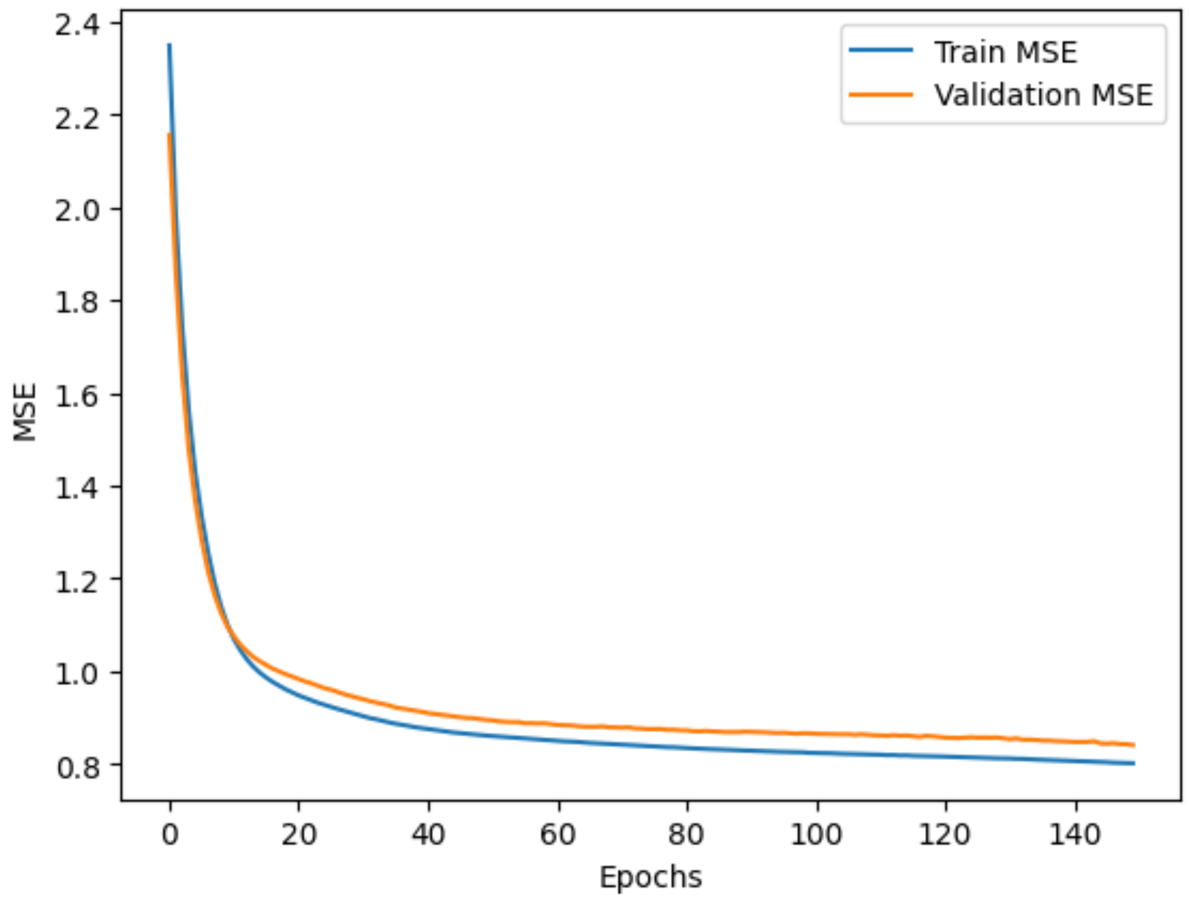
```
Epoch 41/150
32/32 - 0s - 9ms/step - loss: 0.8749 - mse: 0.8749 - val_loss: 0.9096 - val
Epoch 42/150
32/32 - 0s - 5ms/step - loss: 0.8731 - mse: 0.8731 - val_loss: 0.9076 - val
Epoch 43/150
32/32 - 0s - 10ms/step - loss: 0.8714 - mse: 0.8714 - val_loss: 0.9061 - va
Epoch 44/150
32/32 - 0s - 9ms/step - loss: 0.8696 - mse: 0.8696 - val_loss: 0.9044 - val
Epoch 45/150
32/32 - 0s - 5ms/step - loss: 0.8677 - mse: 0.8677 - val_loss: 0.9023 - val
Epoch 46/150
32/32 - 0s - 5ms/step - loss: 0.8663 - mse: 0.8663 - val_loss: 0.9010 - val
Epoch 47/150
32/32 - 0s - 6ms/step - loss: 0.8653 - mse: 0.8653 - val_loss: 0.8989 - val
Epoch 48/150
32/32 - 0s - 6ms/step - loss: 0.8636 - mse: 0.8636 - val_loss: 0.8988 - val
Epoch 49/150
32/32 - 0s - 9ms/step - loss: 0.8624 - mse: 0.8624 - val_loss: 0.8966 - val
Epoch 50/150
32/32 - 0s - 9ms/step - loss: 0.8611 - mse: 0.8611 - val_loss: 0.8948 - val
Epoch 51/150
32/32 - 0s - 5ms/step - loss: 0.8601 - mse: 0.8601 - val_loss: 0.8941 - val
Epoch 52/150
32/32 - 0s - 9ms/step - loss: 0.8588 - mse: 0.8588 - val_loss: 0.8918 - val
Epoch 53/150
32/32 - 0s - 5ms/step - loss: 0.8577 - mse: 0.8577 - val_loss: 0.8907 - val
Epoch 54/150
32/32 - 0s - 5ms/step - loss: 0.8567 - mse: 0.8567 - val_loss: 0.8902 - val
Epoch 55/150
32/32 - 0s - 9ms/step - loss: 0.8556 - mse: 0.8556 - val_loss: 0.8903 - val
Epoch 56/150
32/32 - 0s - 5ms/step - loss: 0.8546 - mse: 0.8546 - val_loss: 0.8876 - val
Epoch 57/150
32/32 - 0s - 5ms/step - loss: 0.8536 - mse: 0.8536 - val_loss: 0.8879 - val
Epoch 58/150
32/32 - 0s - 9ms/step - loss: 0.8527 - mse: 0.8527 - val_loss: 0.8872 - val
Epoch 59/150
32/32 - 0s - 9ms/step - loss: 0.8520 - mse: 0.8520 - val_loss: 0.8875 - val
Epoch 60/150
32/32 - 0s - 9ms/step - loss: 0.8510 - mse: 0.8510 - val_loss: 0.8856 - val
Epoch 61/150
32/32 - 0s - 9ms/step - loss: 0.8497 - mse: 0.8497 - val_loss: 0.8841 - val
Epoch 62/150
32/32 - 0s - 10ms/step - loss: 0.8488 - mse: 0.8488 - val_loss: 0.8838 - va
Epoch 63/150
32/32 - 0s - 9ms/step - loss: 0.8484 - mse: 0.8484 - val_loss: 0.8832 - val
Epoch 64/150
32/32 - 0s - 6ms/step - loss: 0.8477 - mse: 0.8477 - val_loss: 0.8817 - val
Epoch 65/150
32/32 - 0s - 5ms/step - loss: 0.8463 - mse: 0.8463 - val_loss: 0.8804 - val
Epoch 66/150
32/32 - 0s - 11ms/step - loss: 0.8453 - mse: 0.8453 - val_loss: 0.8804 - va
Epoch 67/150
32/32 - 0s - 7ms/step - loss: 0.8447 - mse: 0.8447 - val_loss: 0.8801 - val
```

```
Epoch 68/150
32/32 - 0s - 9ms/step - loss: 0.8439 - mse: 0.8439 - val_loss: 0.8810 - val
Epoch 69/150
32/32 - 0s - 12ms/step - loss: 0.8430 - mse: 0.8430 - val_loss: 0.8791 - va
Epoch 70/150
32/32 - 1s - 17ms/step - loss: 0.8424 - mse: 0.8424 - val_loss: 0.8788 - va
Epoch 71/150
32/32 - 0s - 10ms/step - loss: 0.8416 - mse: 0.8416 - val_loss: 0.8782 - va
Epoch 72/150
32/32 - 0s - 8ms/step - loss: 0.8409 - mse: 0.8409 - val_loss: 0.8790 - val
Epoch 73/150
32/32 - 0s - 7ms/step - loss: 0.8400 - mse: 0.8400 - val_loss: 0.8766 - val
Epoch 74/150
32/32 - 0s - 10ms/step - loss: 0.8396 - mse: 0.8396 - val_loss: 0.8762 - va
Epoch 75/150
32/32 - 0s - 5ms/step - loss: 0.8384 - mse: 0.8384 - val_loss: 0.8750 - val
Epoch 76/150
32/32 - 0s - 9ms/step - loss: 0.8377 - mse: 0.8377 - val_loss: 0.8754 - val
Epoch 77/150
32/32 - 0s - 9ms/step - loss: 0.8371 - mse: 0.8371 - val_loss: 0.8753 - val
Epoch 78/150
32/32 - 0s - 9ms/step - loss: 0.8362 - mse: 0.8362 - val_loss: 0.8737 - val
Epoch 79/150
32/32 - 0s - 10ms/step - loss: 0.8359 - mse: 0.8359 - val_loss: 0.8737 - va
Epoch 80/150
32/32 - 0s - 5ms/step - loss: 0.8350 - mse: 0.8350 - val_loss: 0.8725 - val
Epoch 81/150
32/32 - 0s - 9ms/step - loss: 0.8341 - mse: 0.8341 - val_loss: 0.8724 - val
Epoch 82/150
32/32 - 0s - 5ms/step - loss: 0.8336 - mse: 0.8336 - val_loss: 0.8707 - val
Epoch 83/150
32/32 - 0s - 6ms/step - loss: 0.8328 - mse: 0.8328 - val_loss: 0.8703 - val
Epoch 84/150
32/32 - 0s - 9ms/step - loss: 0.8325 - mse: 0.8325 - val_loss: 0.8711 - val
Epoch 85/150
32/32 - 0s - 10ms/step - loss: 0.8317 - mse: 0.8317 - val_loss: 0.8701 - va
Epoch 86/150
32/32 - 0s - 9ms/step - loss: 0.8313 - mse: 0.8313 - val_loss: 0.8697 - val
Epoch 87/150
32/32 - 0s - 9ms/step - loss: 0.8307 - mse: 0.8307 - val_loss: 0.8687 - val
Epoch 88/150
32/32 - 0s - 5ms/step - loss: 0.8301 - mse: 0.8301 - val_loss: 0.8684 - val
Epoch 89/150
32/32 - 0s - 5ms/step - loss: 0.8294 - mse: 0.8294 - val_loss: 0.8686 - val
Epoch 90/150
32/32 - 0s - 10ms/step - loss: 0.8290 - mse: 0.8290 - val_loss: 0.8696 - va
Epoch 91/150
32/32 - 0s - 9ms/step - loss: 0.8287 - mse: 0.8287 - val_loss: 0.8692 - val
Epoch 92/150
32/32 - 0s - 9ms/step - loss: 0.8280 - mse: 0.8280 - val_loss: 0.8689 - val
Epoch 93/150
32/32 - 0s - 9ms/step - loss: 0.8273 - mse: 0.8273 - val_loss: 0.8683 - val
Epoch 94/150
32/32 - 0s - 9ms/step - loss: 0.8271 - mse: 0.8271 - val_loss: 0.8674 - val
```

```
Epoch 95/150
32/32 - 0s - 10ms/step - loss: 0.8264 - mse: 0.8264 - val_loss: 0.8666 - va
Epoch 96/150
32/32 - 0s - 8ms/step - loss: 0.8261 - mse: 0.8261 - val_loss: 0.8672 - val
Epoch 97/150
32/32 - 0s - 5ms/step - loss: 0.8255 - mse: 0.8255 - val_loss: 0.8663 - val
Epoch 98/150
32/32 - 0s - 9ms/step - loss: 0.8253 - mse: 0.8253 - val_loss: 0.8651 - val
Epoch 99/150
32/32 - 0s - 9ms/step - loss: 0.8250 - mse: 0.8250 - val_loss: 0.8662 - val
Epoch 100/150
32/32 - 0s - 5ms/step - loss: 0.8239 - mse: 0.8239 - val_loss: 0.8659 - val
Epoch 101/150
32/32 - 0s - 9ms/step - loss: 0.8239 - mse: 0.8239 - val_loss: 0.8650 - val
Epoch 102/150
32/32 - 0s - 10ms/step - loss: 0.8233 - mse: 0.8233 - val_loss: 0.8646 - va
Epoch 103/150
32/32 - 0s - 12ms/step - loss: 0.8232 - mse: 0.8232 - val_loss: 0.8647 - va
Epoch 104/150
32/32 - 0s - 7ms/step - loss: 0.8225 - mse: 0.8225 - val_loss: 0.8639 - val
Epoch 105/150
32/32 - 0s - 9ms/step - loss: 0.8220 - mse: 0.8220 - val_loss: 0.8640 - val
Epoch 106/150
32/32 - 0s - 5ms/step - loss: 0.8216 - mse: 0.8216 - val_loss: 0.8640 - val
Epoch 107/150
32/32 - 0s - 10ms/step - loss: 0.8210 - mse: 0.8210 - val_loss: 0.8628 - va
Epoch 108/150
32/32 - 0s - 9ms/step - loss: 0.8208 - mse: 0.8208 - val_loss: 0.8641 - val
Epoch 109/150
32/32 - 0s - 13ms/step - loss: 0.8203 - mse: 0.8203 - val_loss: 0.8627 - va
Epoch 110/150
32/32 - 0s - 8ms/step - loss: 0.8201 - mse: 0.8201 - val_loss: 0.8617 - val
Epoch 111/150
32/32 - 0s - 9ms/step - loss: 0.8199 - mse: 0.8199 - val_loss: 0.8615 - val
Epoch 112/150
32/32 - 0s - 9ms/step - loss: 0.8190 - mse: 0.8190 - val_loss: 0.8607 - val
Epoch 113/150
32/32 - 0s - 12ms/step - loss: 0.8192 - mse: 0.8192 - val_loss: 0.8620 - va
Epoch 114/150
32/32 - 1s - 17ms/step - loss: 0.8183 - mse: 0.8183 - val_loss: 0.8608 - va
Epoch 115/150
32/32 - 0s - 7ms/step - loss: 0.8184 - mse: 0.8184 - val_loss: 0.8614 - val
Epoch 116/150
32/32 - 0s - 8ms/step - loss: 0.8175 - mse: 0.8175 - val_loss: 0.8599 - val
Epoch 117/150
32/32 - 0s - 9ms/step - loss: 0.8173 - mse: 0.8173 - val_loss: 0.8583 - val
Epoch 118/150
32/32 - 0s - 5ms/step - loss: 0.8167 - mse: 0.8167 - val_loss: 0.8605 - val
Epoch 119/150
32/32 - 0s - 9ms/step - loss: 0.8167 - mse: 0.8167 - val_loss: 0.8598 - val
Epoch 120/150
32/32 - 0s - 9ms/step - loss: 0.8163 - mse: 0.8163 - val_loss: 0.8584 - val
Epoch 121/150
32/32 - 0s - 5ms/step - loss: 0.8158 - mse: 0.8158 - val_loss: 0.8569 - val
```

```
Epoch 122/150
32/32 - 0s - 9ms/step - loss: 0.8155 - mse: 0.8155 - val_loss: 0.8566 - val
Epoch 123/150
32/32 - 0s - 10ms/step - loss: 0.8151 - mse: 0.8151 - val_loss: 0.8559 - va
Epoch 124/150
32/32 - 0s - 9ms/step - loss: 0.8149 - mse: 0.8149 - val_loss: 0.8566 - val
Epoch 125/150
32/32 - 0s - 10ms/step - loss: 0.8143 - mse: 0.8143 - val_loss: 0.8576 - va
Epoch 126/150
32/32 - 0s - 9ms/step - loss: 0.8138 - mse: 0.8138 - val_loss: 0.8566 - val
Epoch 127/150
32/32 - 0s - 9ms/step - loss: 0.8132 - mse: 0.8132 - val_loss: 0.8569 - val
Epoch 128/150
32/32 - 0s - 9ms/step - loss: 0.8128 - mse: 0.8128 - val_loss: 0.8566 - val
Epoch 129/150
32/32 - 0s - 10ms/step - loss: 0.8122 - mse: 0.8122 - val_loss: 0.8570 - va
Epoch 130/150
32/32 - 0s - 5ms/step - loss: 0.8122 - mse: 0.8122 - val_loss: 0.8554 - val
Epoch 131/150
32/32 - 0s - 9ms/step - loss: 0.8118 - mse: 0.8118 - val_loss: 0.8534 - val
Epoch 132/150
32/32 - 0s - 9ms/step - loss: 0.8110 - mse: 0.8110 - val_loss: 0.8550 - val
Epoch 133/150
32/32 - 0s - 10ms/step - loss: 0.8109 - mse: 0.8109 - val_loss: 0.8523 - va
Epoch 134/150
32/32 - 0s - 6ms/step - loss: 0.8100 - mse: 0.8100 - val_loss: 0.8524 - val
Epoch 135/150
32/32 - 0s - 5ms/step - loss: 0.8095 - mse: 0.8095 - val_loss: 0.8517 - val
Epoch 136/150
32/32 - 0s - 10ms/step - loss: 0.8090 - mse: 0.8090 - val_loss: 0.8505 - va
Epoch 137/150
32/32 - 0s - 5ms/step - loss: 0.8087 - mse: 0.8087 - val_loss: 0.8504 - val
Epoch 138/150
32/32 - 0s - 10ms/step - loss: 0.8080 - mse: 0.8080 - val_loss: 0.8494 - va
Epoch 139/150
32/32 - 0s - 9ms/step - loss: 0.8077 - mse: 0.8077 - val_loss: 0.8490 - val
Epoch 140/150
32/32 - 0s - 5ms/step - loss: 0.8074 - mse: 0.8074 - val_loss: 0.8483 - val
Epoch 141/150
32/32 - 0s - 9ms/step - loss: 0.8068 - mse: 0.8068 - val_loss: 0.8474 - val
Epoch 142/150
32/32 - 0s - 9ms/step - loss: 0.8060 - mse: 0.8060 - val_loss: 0.8475 - val
Epoch 143/150
32/32 - 0s - 5ms/step - loss: 0.8053 - mse: 0.8053 - val_loss: 0.8475 - val
Epoch 144/150
32/32 - 0s - 9ms/step - loss: 0.8050 - mse: 0.8050 - val_loss: 0.8491 - val
Epoch 145/150
32/32 - 0s - 5ms/step - loss: 0.8043 - mse: 0.8043 - val_loss: 0.8444 - val
Epoch 146/150
32/32 - 0s - 10ms/step - loss: 0.8037 - mse: 0.8037 - val_loss: 0.8437 - va
Epoch 147/150
32/32 - 0s - 10ms/step - loss: 0.8028 - mse: 0.8028 - val_loss: 0.8446 - va
Epoch 148/150
32/32 - 0s - 9ms/step - loss: 0.8026 - mse: 0.8026 - val_loss: 0.8429 - val
```

```
Epoch 149/150
32/32 - 0s - 5ms/step - loss: 0.8019 - mse: 0.8019 - val_loss: 0.8428 - val
Epoch 150/150
32/32 - 0s - 9ms/step - loss: 0.8015 - mse: 0.8015 - val_loss: 0.8408 - val
```



```
# Model with Different Epochs - 1
model_9 = create_ffnn(activation_functions=['tanh', 'leaky_relu', 'linear'])
history_9 = train_model(model_9,learning_rate=0.01,epochs=225)

# Plot Training & Validation Loss
plt.plot(history_9.history['loss'], label='Train MSE')
plt.plot(history_9.history['val_loss'], label='Validation MSE')
plt.xlabel('Epochs')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

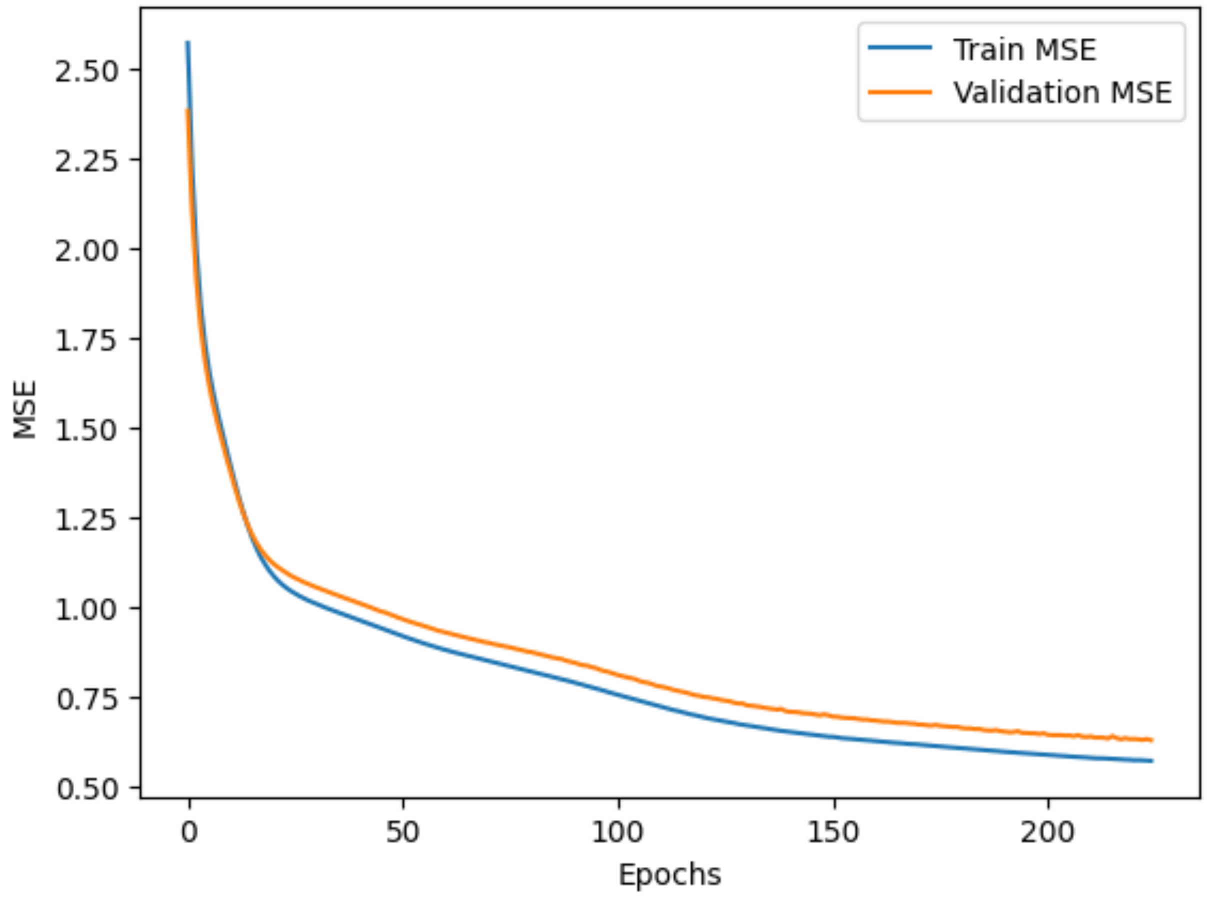
```
Epoch 1/225
32/32 - 1s - 26ms/step - loss: 2.5697 - mse: 2.5697 - val_loss: 2.3809 - va
Epoch 2/225
```

```
32/32 - 0s - 5ms/step - loss: 2.2276 - mse: 2.2276 - val_loss: 2.1017 - val
Epoch 3/225
32/32 - 0s - 10ms/step - loss: 1.9992 - mse: 1.9992 - val_loss: 1.9088 - va
Epoch 4/225
32/32 - 0s - 5ms/step - loss: 1.8414 - mse: 1.8414 - val_loss: 1.7773 - val
Epoch 5/225
32/32 - 0s - 5ms/step - loss: 1.7334 - mse: 1.7334 - val_loss: 1.6842 - val
Epoch 6/225
32/32 - 0s - 5ms/step - loss: 1.6543 - mse: 1.6543 - val_loss: 1.6144 - val
Epoch 7/225
32/32 - 0s - 5ms/step - loss: 1.5912 - mse: 1.5912 - val_loss: 1.5567 - val
Epoch 8/225
32/32 - 0s - 10ms/step - loss: 1.5374 - mse: 1.5374 - val_loss: 1.5061 - va
Epoch 9/225
32/32 - 0s - 9ms/step - loss: 1.4870 - mse: 1.4870 - val_loss: 1.4596 - val
Epoch 10/225
32/32 - 0s - 9ms/step - loss: 1.4380 - mse: 1.4380 - val_loss: 1.4145 - val
Epoch 11/225
32/32 - 0s - 6ms/step - loss: 1.3900 - mse: 1.3900 - val_loss: 1.3721 - val
Epoch 12/225
32/32 - 0s - 9ms/step - loss: 1.3438 - mse: 1.3438 - val_loss: 1.3299 - val
Epoch 13/225
32/32 - 0s - 5ms/step - loss: 1.3002 - mse: 1.3002 - val_loss: 1.2914 - val
Epoch 14/225
32/32 - 0s - 10ms/step - loss: 1.2597 - mse: 1.2597 - val_loss: 1.2562 - va
Epoch 15/225
32/32 - 0s - 9ms/step - loss: 1.2227 - mse: 1.2227 - val_loss: 1.2257 - val
Epoch 16/225
32/32 - 0s - 10ms/step - loss: 1.1901 - mse: 1.1901 - val_loss: 1.1992 - va
Epoch 17/225
32/32 - 0s - 5ms/step - loss: 1.1617 - mse: 1.1617 - val_loss: 1.1772 - val
Epoch 18/225
32/32 - 0s - 6ms/step - loss: 1.1372 - mse: 1.1372 - val_loss: 1.1582 - val
Epoch 19/225
32/32 - 0s - 9ms/step - loss: 1.1173 - mse: 1.1173 - val_loss: 1.1429 - val
Epoch 20/225
32/32 - 0s - 6ms/step - loss: 1.0995 - mse: 1.0995 - val_loss: 1.1296 - val
Epoch 21/225
32/32 - 0s - 9ms/step - loss: 1.0846 - mse: 1.0846 - val_loss: 1.1186 - val
Epoch 22/225
32/32 - 0s - 9ms/step - loss: 1.0718 - mse: 1.0718 - val_loss: 1.1090 - val
Epoch 23/225
32/32 - 0s - 14ms/step - loss: 1.0607 - mse: 1.0607 - val_loss: 1.1009 - va
Epoch 24/225
32/32 - 1s - 16ms/step - loss: 1.0513 - mse: 1.0513 - val_loss: 1.0930 - va
Epoch 25/225
32/32 - 0s - 9ms/step - loss: 1.0429 - mse: 1.0429 - val_loss: 1.0854 - val
Epoch 26/225
32/32 - 0s - 12ms/step - loss: 1.0356 - mse: 1.0356 - val_loss: 1.0793 - va
Epoch 27/225
32/32 - 0s - 10ms/step - loss: 1.0288 - mse: 1.0288 - val_loss: 1.0733 - va
Epoch 28/225
32/32 - 0s - 15ms/step - loss: 1.0226 - mse: 1.0226 - val_loss: 1.0678 - va
Epoch 29/225
```

```
32/32 - 0s - 6ms/step - loss: 1.0169 - mse: 1.0169 - val_loss: 1.0627 - val
Epoch 30/225
32/32 - 0s - 9ms/step - loss: 1.0118 - mse: 1.0118 - val_loss: 1.0579 - val
Epoch 31/225
32/32 - 0s - 9ms/step - loss: 1.0065 - mse: 1.0065 - val_loss: 1.0529 - val
Epoch 32/225
32/32 - 0s - 6ms/step - loss: 1.0018 - mse: 1.0018 - val_loss: 1.0482 - val
Epoch 33/225
32/32 - 0s - 8ms/step - loss: 0.9970 - mse: 0.9970 - val_loss: 1.0436 - val
Epoch 34/225
32/32 - 0s - 10ms/step - loss: 0.9923 - mse: 0.9923 - val_loss: 1.0395 - va
Epoch 35/225
32/32 - 0s - 9ms/step - loss: 0.9880 - mse: 0.9880 - val_loss: 1.0349 - val
Epoch 36/225
32/32 - 0s - 9ms/step - loss: 0.9833 - mse: 0.9833 - val_loss: 1.0303 - val
Epoch 37/225
32/32 - 0s - 9ms/step - loss: 0.9790 - mse: 0.9790 - val_loss: 1.0260 - val
Epoch 38/225
32/32 - 0s - 5ms/step - loss: 0.9746 - mse: 0.9746 - val_loss: 1.0216 - val
Epoch 39/225
32/32 - 0s - 5ms/step - loss: 0.9704 - mse: 0.9704 - val_loss: 1.0173 - val
Epoch 40/225
32/32 - 0s - 9ms/step - loss: 0.9656 - mse: 0.9656 - val_loss: 1.0134 - val
Epoch 41/225
32/32 - 0s - 5ms/step - loss: 0.9614 - mse: 0.9614 - val_loss: 1.0089 - val
Epoch 42/225
32/32 - 0s - 5ms/step - loss: 0.9570 - mse: 0.9570 - val_loss: 1.0048 - val
Epoch 43/225
32/32 - 0s - 10ms/step - loss: 0.9526 - mse: 0.9526 - val_loss: 0.9998 - va
Epoch 44/225
32/32 - 0s - 5ms/step - loss: 0.9481 - mse: 0.9481 - val_loss: 0.9954 - val
Epoch 45/225
32/32 - 0s - 10ms/step - loss: 0.9439 - mse: 0.9439 - val_loss: 0.9909 - va
Epoch 46/225
32/32 - 0s - 5ms/step - loss: 0.9392 - mse: 0.9392 - val_loss: 0.9859 - val
Epoch 47/225
32/32 - 0s - 9ms/step - loss: 0.9346 - mse: 0.9346 - val_loss: 0.9830 - val
Epoch 48/225
32/32 - 0s - 10ms/step - loss: 0.9306 - mse: 0.9306 - val_loss: 0.9780 - va
Epoch 49/225
32/32 - 0s - 5ms/step - loss: 0.9261 - mse: 0.9261 - val_loss: 0.9737 - val
Epoch 50/225
32/32 - 0s - 6ms/step - loss: 0.9216 - mse: 0.9216 - val_loss: 0.9687 - val
Epoch 51/225
32/32 - 0s - 9ms/step - loss: 0.9173 - mse: 0.9173 - val_loss: 0.9649 - val
Epoch 52/225
32/32 - 0s - 10ms/step - loss: 0.9130 - mse: 0.9130 - val_loss: 0.9603 - va
Epoch 53/225
32/32 - 0s - 10ms/step - loss: 0.9088 - mse: 0.9088 - val_loss: 0.9566 - va
Epoch 54/225
32/32 - 0s - 5ms/step - loss: 0.9048 - mse: 0.9048 - val_loss: 0.9536 - val
Epoch 55/225
32/32 - 0s - 9ms/step - loss: 0.9008 - mse: 0.9008 - val_loss: 0.9488 - val
Epoch 56/225
```

```
32/32 - 0s - 9ms/step - loss: 0.8968 - mse: 0.8968 - val_loss: 0.9452 - val
Epoch 57/225
32/32 - 0s - 9ms/step - loss: 0.8930 - mse: 0.8930 - val_loss: 0.9417 - val
Epoch 58/225
32/32 - 0s - 9ms/step - loss: 0.8891 - mse: 0.8891 - val_loss: 0.9376 - val
Epoch 59/225
32/32 - 0s - 9ms/step - loss: 0.8859 - mse: 0.8859 - val_loss: 0.9335 - val
Epoch 60/225
32/32 - 0s - 5ms/step - loss: 0.8822 - mse: 0.8822 - val_loss: 0.9303 - val
Epoch 61/225
32/32 - 0s - 5ms/step - loss: 0.8787 - mse: 0.8787 - val_loss: 0.9271 - val
Epoch 62/225
32/32 - 0s - 9ms/step - loss: 0.8756 - mse: 0.8756 - val_loss: 0.9239 - val
Epoch 63/225
32/32 - 0s - 10ms/step - loss: 0.8722 - mse: 0.8722 - val_loss: 0.9213 - va
Epoch 64/225
32/32 - 0s - 9ms/step - loss: 0.8692 - mse: 0.8692 - val_loss: 0.9183 - val
Epoch 65/225
32/32 - 0s - 5ms/step - loss: 0.8659 - mse: 0.8659 - val_loss: 0.9155 - val
Epoch 66/225
32/32 - 0s - 13ms/step - loss: 0.8630 - mse: 0.8630 - val_loss: 0.9122 - va
Epoch 67/225
32/32 - 0s - 10ms/step - loss: 0.8599 - mse: 0.8599 - val_loss: 0.9094 - va
Epoch 68/225
32/32 - 1s - 17ms/step - loss: 0.8571 - mse: 0.8571 - val_loss: 0.9066 - va
Epoch 69/225
32/32 - 0s - 12ms/step - loss: 0.8539 - mse: 0.8539 - val_loss: 0.9040 - va
Epoch 70/225
32/32 - 1s - 17ms/step - loss: 0.8509 - mse: 0.8509 - val_loss: 0.9007 - va
Epoch 71/225
32/32 - 0s - 8ms/step - loss: 0.8482 - mse: 0.8482 - val_loss: 0.8988 - val
Epoch 72/225
32/32 - 0s - 5ms/step - loss: 0.8456 - mse: 0.8456 - val_loss: 0.8956 - val
Epoch 73/225
32/32 - 0s - 9ms/step - loss: 0.8424 - mse: 0.8424 - val_loss: 0.8933 - val
Epoch 74/225
32/32 - 0s - 9ms/step - loss: 0.8397 - mse: 0.8397 - val_loss: 0.8903 - val
Epoch 75/225
32/32 - 0s - 5ms/step - loss: 0.8366 - mse: 0.8366 - val_loss: 0.8880 - val
Epoch 76/225
32/32 - 0s - 9ms/step - loss: 0.8339 - mse: 0.8339 - val_loss: 0.8859 - val
Epoch 77/225
32/32 - 0s - 5ms/step - loss: 0.8311 - mse: 0.8311 - val_loss: 0.8829 - val
Epoch 78/225
32/32 - 0s - 10ms/step - loss: 0.8279 - mse: 0.8279 - val_loss: 0.8802 - va
Epoch 79/225
32/32 - 0s - 9ms/step - loss: 0.8252 - mse: 0.8252 - val_loss: 0.8772 - val
Epoch 80/225
32/32 - 0s - 9ms/step - loss: 0.8219 - mse: 0.8219 - val_loss: 0.8744 - val
Epoch 81/225
32/32 - 0s - 10ms/step - loss: 0.8193 - mse: 0.8193 - val_loss: 0.8726 - va
Epoch 82/225
32/32 - 0s - 9ms/step - loss: 0.8161 - mse: 0.8161 - val_loss: 0.8694 - val
Epoch 83/225
Epoch 83/225
32/32 - 0s - 9ms/step - loss: 0.8131 - mse: 0.8131 - val_loss: 0.8663 - val
Epoch 84/225
32/32 - 0s - 9ms/step - loss: 0.8102 - mse: 0.8102 - val_loss: 0.8635 - val
Epoch 85/225
32/32 - 0s - 9ms/step - loss: 0.8071 - mse: 0.8071 - val_loss: 0.8606 - val
Epoch 86/225
32/32 - 0s - 9ms/step - loss: 0.8040 - mse: 0.8040 - val_loss: 0.8576 - val
Epoch 87/225
32/32 - 0s - 10ms/step - loss: 0.8009 - mse: 0.8009 - val_loss: 0.8556 - va
Epoch 88/225
32/32 - 0s - 9ms/step - loss: 0.7976 - mse: 0.7976 - val_loss: 0.8534 - val
Epoch 89/225
32/32 - 0s - 10ms/step - loss: 0.7946 - mse: 0.7946 - val_loss: 0.8490 - va
Epoch 90/225
32/32 - 0s - 5ms/step - loss: 0.7913 - mse: 0.7913 - val_loss: 0.8463 - val
Epoch 91/225
32/32 - 0s - 9ms/step - loss: 0.7878 - mse: 0.7878 - val_loss: 0.8432 - val
Epoch 92/225
32/32 - 0s - 10ms/step - loss: 0.7849 - mse: 0.7849 - val_loss: 0.8393 - va
Epoch 93/225
32/32 - 0s - 9ms/step - loss: 0.7814 - mse: 0.7814 - val_loss: 0.8362 - val
Epoch 94/225
32/32 - 0s - 9ms/step - loss: 0.7780 - mse: 0.7780 - val_loss: 0.8351 - val
Epoch 95/225
32/32 - 0s - 9ms/step - loss: 0.7745 - mse: 0.7745 - val_loss: 0.8307 - val
Epoch 96/225
32/32 - 0s - 10ms/step - loss: 0.7715 - mse: 0.7715 - val_loss: 0.8284 - va
Epoch 97/225
32/32 - 0s - 5ms/step - loss: 0.7680 - mse: 0.7680 - val_loss: 0.8229 - val
Epoch 98/225
32/32 - 0s - 10ms/step - loss: 0.7643 - mse: 0.7643 - val_loss: 0.8199 - va
Epoch 99/225
32/32 - 0s - 5ms/step - loss: 0.7609 - mse: 0.7609 - val_loss: 0.8171 - val
Epoch 100/225
32/32 - 0s - 10ms/step - loss: 0.7573 - mse: 0.7573 - val_loss: 0.8130 - va
Epoch 101/225
32/32 - 0s - 9ms/step - loss: 0.7540 - mse: 0.7540 - val_loss: 0.8092 - val
Epoch 102/225
32/32 - 0s - 9ms/step - loss: 0.7506 - mse: 0.7506 - val_loss: 0.8060 - val
Epoch 103/225
32/32 - 0s - 9ms/step - loss: 0.7472 - mse: 0.7472 - val_loss: 0.8032 - val
Epoch 104/225
32/32 - 0s - 9ms/step - loss: 0.7437 - mse: 0.7437 - val_loss: 0.8007 - val
Epoch 105/225
32/32 - 0s - 6ms/step - loss: 0.7403 - mse: 0.7403 - val_loss: 0.7979 - val
Epoch 106/225
32/32 - 0s - 12ms/step - loss: 0.7372 - mse: 0.7372 - val_loss: 0.7928 - va
Epoch 107/225
32/32 - 0s - 11ms/step - loss: 0.7337 - mse: 0.7337 - val_loss: 0.7895 - va
Epoch 108/225
32/32 - 1s - 17ms/step - loss: 0.7302 - mse: 0.7302 - val_loss: 0.7873 - va
Epoch 109/225
32/32 - 0s - 12ms/step - loss: 0.7271 - mse: 0.7271 - val_loss: 0.7836 - va
Epoch 110/225
Epoch 110/225
32/32 - 0s - 10ms/step - loss: 0.7241 - mse: 0.7241 - val_loss: 0.7791 - va
Epoch 111/225
32/32 - 0s - 15ms/step - loss: 0.7207 - mse: 0.7207 - val_loss: 0.7771 - va
Epoch 112/225
32/32 - 0s - 10ms/step - loss: 0.7175 - mse: 0.7175 - val_loss: 0.7743 - va
Epoch 113/225
32/32 - 0s - 9ms/step - loss: 0.7148 - mse: 0.7148 - val_loss: 0.7714 - val
Epoch 114/225
32/32 - 0s - 9ms/step - loss: 0.7114 - mse: 0.7114 - val_loss: 0.7676 - val
Epoch 115/225
32/32 - 0s - 5ms/step - loss: 0.7089 - mse: 0.7089 - val_loss: 0.7651 - val
Epoch 116/225
32/32 - 0s - 10ms/step - loss: 0.7055 - mse: 0.7055 - val_loss: 0.7632 - va
Epoch 117/225
32/32 - 0s - 5ms/step - loss: 0.7028 - mse: 0.7028 - val_loss: 0.7598 - val
Epoch 118/225
32/32 - 0s - 10ms/step - loss: 0.6992 - mse: 0.6992 - val_loss: 0.7560 - va
Epoch 119/225
32/32 - 0s - 9ms/step - loss: 0.6971 - mse: 0.6971 - val_loss: 0.7527 - val
Epoch 120/225
32/32 - 0s - 9ms/step - loss: 0.6941 - mse: 0.6941 - val_loss: 0.7510 - val
Epoch 121/225
32/32 - 0s - 5ms/step - loss: 0.6916 - mse: 0.6916 - val_loss: 0.7479 - val
Epoch 122/225
32/32 - 0s - 5ms/step - loss: 0.6886 - mse: 0.6886 - val_loss: 0.7471 - val
Epoch 123/225
32/32 - 0s - 5ms/step - loss: 0.6864 - mse: 0.6864 - val_loss: 0.7442 - val
Epoch 124/225
32/32 - 0s - 6ms/step - loss: 0.6842 - mse: 0.6842 - val_loss: 0.7427 - val
Epoch 125/225
32/32 - 0s - 9ms/step - loss: 0.6817 - mse: 0.6817 - val_loss: 0.7396 - val
Epoch 126/225
32/32 - 0s - 5ms/step - loss: 0.6794 - mse: 0.6794 - val_loss: 0.7378 - val
Epoch 127/225
32/32 - 0s - 10ms/step - loss: 0.6772 - mse: 0.6772 - val_loss: 0.7361 - va
Epoch 128/225
32/32 - 0s - 6ms/step - loss: 0.6753 - mse: 0.6753 - val_loss: 0.7322 - val
Epoch 129/225
32/32 - 0s - 9ms/step - loss: 0.6726 - mse: 0.6726 - val_loss: 0.7298 - val
Epoch 130/225
32/32 - 0s - 9ms/step - loss: 0.6707 - mse: 0.6707 - val_loss: 0.7297 - val
Epoch 131/225
32/32 - 0s - 5ms/step - loss: 0.6689 - mse: 0.6689 - val_loss: 0.7249 - val
Epoch 132/225
32/32 - 0s - 6ms/step - loss: 0.6668 - mse: 0.6668 - val_loss: 0.7237 - val
Epoch 133/225
32/32 - 0s - 9ms/step - loss: 0.6646 - mse: 0.6646 - val_loss: 0.7222 - val
Epoch 134/225
32/32 - 0s - 5ms/step - loss: 0.6624 - mse: 0.6624 - val_loss: 0.7206 - val
Epoch 135/225
32/32 - 0s - 9ms/step - loss: 0.6607 - mse: 0.6607 - val_loss: 0.7182 - val
Epoch 136/225
32/32 - 0s - 10ms/step - loss: 0.6586 - mse: 0.6586 - val_loss: 0.7165 - va
Epoch 137/225
Epoch 137/225
32/32 - 0s - 9ms/step - loss: 0.6569 - mse: 0.6569 - val_loss: 0.7143 - val
Epoch 138/225
32/32 - 0s - 9ms/step - loss: 0.6552 - mse: 0.6552 - val_loss: 0.7129 - val
Epoch 139/225
32/32 - 0s - 10ms/step - loss: 0.6536 - mse: 0.6536 - val_loss: 0.7147 - va
Epoch 140/225
32/32 - 0s - 9ms/step - loss: 0.6521 - mse: 0.6521 - val_loss: 0.7087 - val
Epoch 141/225
32/32 - 0s - 9ms/step - loss: 0.6506 - mse: 0.6506 - val_loss: 0.7065 - val
Epoch 142/225
32/32 - 0s - 5ms/step - loss: 0.6487 - mse: 0.6487 - val_loss: 0.7063 - val
Epoch 143/225
32/32 - 0s - 5ms/step - loss: 0.6472 - mse: 0.6472 - val_loss: 0.7053 - val
Epoch 144/225
32/32 - 0s - 10ms/step - loss: 0.6458 - mse: 0.6458 - val_loss: 0.7030 - va
Epoch 145/225
32/32 - 0s - 9ms/step - loss: 0.6444 - mse: 0.6444 - val_loss: 0.7019 - val
Epoch 146/225
32/32 - 0s - 5ms/step - loss: 0.6433 - mse: 0.6433 - val_loss: 0.7014 - val
Epoch 147/225
32/32 - 0s - 10ms/step - loss: 0.6416 - mse: 0.6416 - val_loss: 0.6988 - va
Epoch 148/225
32/32 - 0s - 9ms/step - loss: 0.6406 - mse: 0.6406 - val_loss: 0.6970 - val
Epoch 149/225
32/32 - 0s - 9ms/step - loss: 0.6385 - mse: 0.6385 - val_loss: 0.6995 - val
Epoch 150/225
32/32 - 0s - 9ms/step - loss: 0.6374 - mse: 0.6374 - val_loss: 0.6969 - val
Epoch 151/225
32/32 - 1s - 19ms/step - loss: 0.6370 - mse: 0.6370 - val_loss: 0.6933 - va
Epoch 152/225
32/32 - 1s - 17ms/step - loss: 0.6355 - mse: 0.6355 - val_loss: 0.6928 - va
Epoch 153/225
32/32 - 0s - 10ms/step - loss: 0.6333 - mse: 0.6333 - val_loss: 0.6909 - va
Epoch 154/225
32/32 - 0s - 8ms/step - loss: 0.6326 - mse: 0.6326 - val_loss: 0.6897 - val
Epoch 155/225
32/32 - 0s - 8ms/step - loss: 0.6317 - mse: 0.6317 - val_loss: 0.6893 - val
Epoch 156/225
32/32 - 0s - 9ms/step - loss: 0.6306 - mse: 0.6306 - val_loss: 0.6884 - val
Epoch 157/225
32/32 - 0s - 5ms/step - loss: 0.6296 - mse: 0.6296 - val_loss: 0.6866 - val
Epoch 158/225
32/32 - 0s - 9ms/step - loss: 0.6285 - mse: 0.6285 - val_loss: 0.6849 - val
Epoch 159/225
32/32 - 0s - 5ms/step - loss: 0.6274 - mse: 0.6274 - val_loss: 0.6851 - val
Epoch 160/225
32/32 - 0s - 9ms/step - loss: 0.6262 - mse: 0.6262 - val_loss: 0.6830 - val
Epoch 161/225
32/32 - 0s - 5ms/step - loss: 0.6251 - mse: 0.6251 - val_loss: 0.6821 - val
Epoch 162/225
32/32 - 0s - 5ms/step - loss: 0.6241 - mse: 0.6241 - val_loss: 0.6804 - val
Epoch 163/225
32/32 - 0s - 10ms/step - loss: 0.6231 - mse: 0.6231 - val_loss: 0.6794 - va
Epoch 164/225
Epoch 164/225
32/32 - 0s - 9ms/step - loss: 0.6219 - mse: 0.6219 - val_loss: 0.6793 - val
Epoch 165/225
32/32 - 0s - 5ms/step - loss: 0.6209 - mse: 0.6209 - val_loss: 0.6775 - val
Epoch 166/225
32/32 - 0s - 9ms/step - loss: 0.6203 - mse: 0.6203 - val_loss: 0.6760 - val
Epoch 167/225
32/32 - 0s - 9ms/step - loss: 0.6192 - mse: 0.6192 - val_loss: 0.6759 - val
Epoch 168/225
32/32 - 0s - 5ms/step - loss: 0.6177 - mse: 0.6177 - val_loss: 0.6755 - val
Epoch 169/225
32/32 - 0s - 10ms/step - loss: 0.6168 - mse: 0.6168 - val_loss: 0.6738 - va
Epoch 170/225
32/32 - 0s - 9ms/step - loss: 0.6164 - mse: 0.6164 - val_loss: 0.6733 - val
Epoch 171/225
32/32 - 0s - 9ms/step - loss: 0.6153 - mse: 0.6153 - val_loss: 0.6712 - val
Epoch 172/225
32/32 - 0s - 5ms/step - loss: 0.6139 - mse: 0.6139 - val_loss: 0.6708 - val
Epoch 173/225
32/32 - 0s - 6ms/step - loss: 0.6129 - mse: 0.6129 - val_loss: 0.6691 - val
Epoch 174/225
32/32 - 0s - 9ms/step - loss: 0.6123 - mse: 0.6123 - val_loss: 0.6683 - val
Epoch 175/225
32/32 - 0s - 10ms/step - loss: 0.6110 - mse: 0.6110 - val_loss: 0.6702 - va
Epoch 176/225
32/32 - 0s - 9ms/step - loss: 0.6101 - mse: 0.6101 - val_loss: 0.6675 - val
Epoch 177/225
32/32 - 0s - 9ms/step - loss: 0.6091 - mse: 0.6091 - val_loss: 0.6664 - val
Epoch 178/225
32/32 - 0s - 9ms/step - loss: 0.6082 - mse: 0.6082 - val_loss: 0.6655 - val
Epoch 179/225
32/32 - 0s - 5ms/step - loss: 0.6073 - mse: 0.6073 - val_loss: 0.6645 - val
Epoch 180/225
32/32 - 0s - 5ms/step - loss: 0.6061 - mse: 0.6061 - val_loss: 0.6646 - val
Epoch 181/225
32/32 - 0s - 5ms/step - loss: 0.6056 - mse: 0.6056 - val_loss: 0.6615 - val
Epoch 182/225
32/32 - 0s - 5ms/step - loss: 0.6042 - mse: 0.6042 - val_loss: 0.6604 - val
Epoch 183/225
32/32 - 0s - 5ms/step - loss: 0.6034 - mse: 0.6034 - val_loss: 0.6605 - val
Epoch 184/225
32/32 - 0s - 5ms/step - loss: 0.6021 - mse: 0.6021 - val_loss: 0.6586 - val
Epoch 185/225
32/32 - 0s - 9ms/step - loss: 0.6014 - mse: 0.6014 - val_loss: 0.6590 - val
Epoch 186/225
32/32 - 0s - 9ms/step - loss: 0.6003 - mse: 0.6003 - val_loss: 0.6562 - val
Epoch 187/225
32/32 - 0s - 5ms/step - loss: 0.5992 - mse: 0.5992 - val_loss: 0.6548 - val
Epoch 188/225
32/32 - 0s - 10ms/step - loss: 0.5985 - mse: 0.5985 - val_loss: 0.6542 - va
Epoch 189/225
32/32 - 0s - 9ms/step - loss: 0.5974 - mse: 0.5974 - val_loss: 0.6558 - val
Epoch 190/225
32/32 - 0s - 5ms/step - loss: 0.5963 - mse: 0.5963 - val_loss: 0.6528 - val
Epoch 191/225
Epoch 191/225
32/32 - 0s - 9ms/step - loss: 0.5953 - mse: 0.5953 - val_loss: 0.6515 - val
Epoch 192/225
32/32 - 0s - 10ms/step - loss: 0.5947 - mse: 0.5947 - val_loss: 0.6498 - va
Epoch 193/225
32/32 - 0s - 10ms/step - loss: 0.5935 - mse: 0.5935 - val_loss: 0.6501 - va
Epoch 194/225
32/32 - 0s - 12ms/step - loss: 0.5927 - mse: 0.5927 - val_loss: 0.6527 - va
Epoch 195/225
32/32 - 0s - 10ms/step - loss: 0.5921 - mse: 0.5921 - val_loss: 0.6479 - va
Epoch 196/225
32/32 - 0s - 7ms/step - loss: 0.5910 - mse: 0.5910 - val_loss: 0.6478 - val
Epoch 197/225
32/32 - 0s - 9ms/step - loss: 0.5900 - mse: 0.5900 - val_loss: 0.6467 - val
Epoch 198/225
32/32 - 0s - 9ms/step - loss: 0.5894 - mse: 0.5894 - val_loss: 0.6468 - val
Epoch 199/225
32/32 - 0s - 10ms/step - loss: 0.5885 - mse: 0.5885 - val_loss: 0.6446 - va
Epoch 200/225
32/32 - 0s - 12ms/step - loss: 0.5873 - mse: 0.5873 - val_loss: 0.6467 - va
Epoch 201/225
32/32 - 0s - 15ms/step - loss: 0.5870 - mse: 0.5870 - val_loss: 0.6428 - va
Epoch 202/225
32/32 - 0s - 9ms/step - loss: 0.5859 - mse: 0.5859 - val_loss: 0.6417 - val
Epoch 203/225
32/32 - 0s - 9ms/step - loss: 0.5846 - mse: 0.5846 - val_loss: 0.6417 - val
Epoch 204/225
32/32 - 0s - 9ms/step - loss: 0.5836 - mse: 0.5836 - val_loss: 0.6417 - val
Epoch 205/225
32/32 - 0s - 5ms/step - loss: 0.5832 - mse: 0.5832 - val_loss: 0.6404 - val
Epoch 206/225
32/32 - 0s - 5ms/step - loss: 0.5822 - mse: 0.5822 - val_loss: 0.6410 - val
Epoch 207/225
32/32 - 0s - 5ms/step - loss: 0.5816 - mse: 0.5816 - val_loss: 0.6384 - val
Epoch 208/225
32/32 - 0s - 10ms/step - loss: 0.5810 - mse: 0.5810 - val_loss: 0.6418 - va
Epoch 209/225
32/32 - 0s - 9ms/step - loss: 0.5807 - mse: 0.5807 - val_loss: 0.6376 - val
Epoch 210/225
32/32 - 0s - 9ms/step - loss: 0.5794 - mse: 0.5794 - val_loss: 0.6366 - val
Epoch 211/225
32/32 - 0s - 5ms/step - loss: 0.5784 - mse: 0.5784 - val_loss: 0.6379 - val
Epoch 212/225
32/32 - 0s - 6ms/step - loss: 0.5775 - mse: 0.5775 - val_loss: 0.6349 - val
Epoch 213/225
32/32 - 0s - 9ms/step - loss: 0.5775 - mse: 0.5775 - val_loss: 0.6351 - val
Epoch 214/225
32/32 - 0s - 5ms/step - loss: 0.5770 - mse: 0.5770 - val_loss: 0.6345 - val
Epoch 215/225
32/32 - 0s - 10ms/step - loss: 0.5762 - mse: 0.5762 - val_loss: 0.6322 - va
Epoch 216/225
32/32 - 0s - 5ms/step - loss: 0.5751 - mse: 0.5751 - val_loss: 0.6390 - val
Epoch 217/225
32/32 - 0s - 5ms/step - loss: 0.5746 - mse: 0.5746 - val_loss: 0.6336 - val
```

```
Epoch 218/225
32/32 - 0s - 5ms/step - loss: 0.5745 - mse: 0.5745 - val_loss: 0.6301 - val
Epoch 219/225
32/32 - 0s - 5ms/step - loss: 0.5735 - mse: 0.5735 - val_loss: 0.6333 - val
Epoch 220/225
32/32 - 0s - 10ms/step - loss: 0.5732 - mse: 0.5732 - val_loss: 0.6307 - va
Epoch 221/225
32/32 - 0s - 9ms/step - loss: 0.5718 - mse: 0.5718 - val_loss: 0.6309 - val
Epoch 222/225
32/32 - 0s - 5ms/step - loss: 0.5721 - mse: 0.5721 - val_loss: 0.6301 - val
Epoch 223/225
32/32 - 0s - 10ms/step - loss: 0.5714 - mse: 0.5714 - val_loss: 0.6284 - va
Epoch 224/225
32/32 - 0s - 5ms/step - loss: 0.5712 - mse: 0.5712 - val_loss: 0.6309 - val
Epoch 225/225
32/32 - 0s - 5ms/step - loss: 0.5703 - mse: 0.5703 - val_loss: 0.6277 - val
```



```
print(history_9.history['loss'][-1])
print(model_9.evaluate(X_train,Y_train,verbose=0))
```

```
0.5703062415122986
[0.5679970383644104, 0.5679970383644104]
```

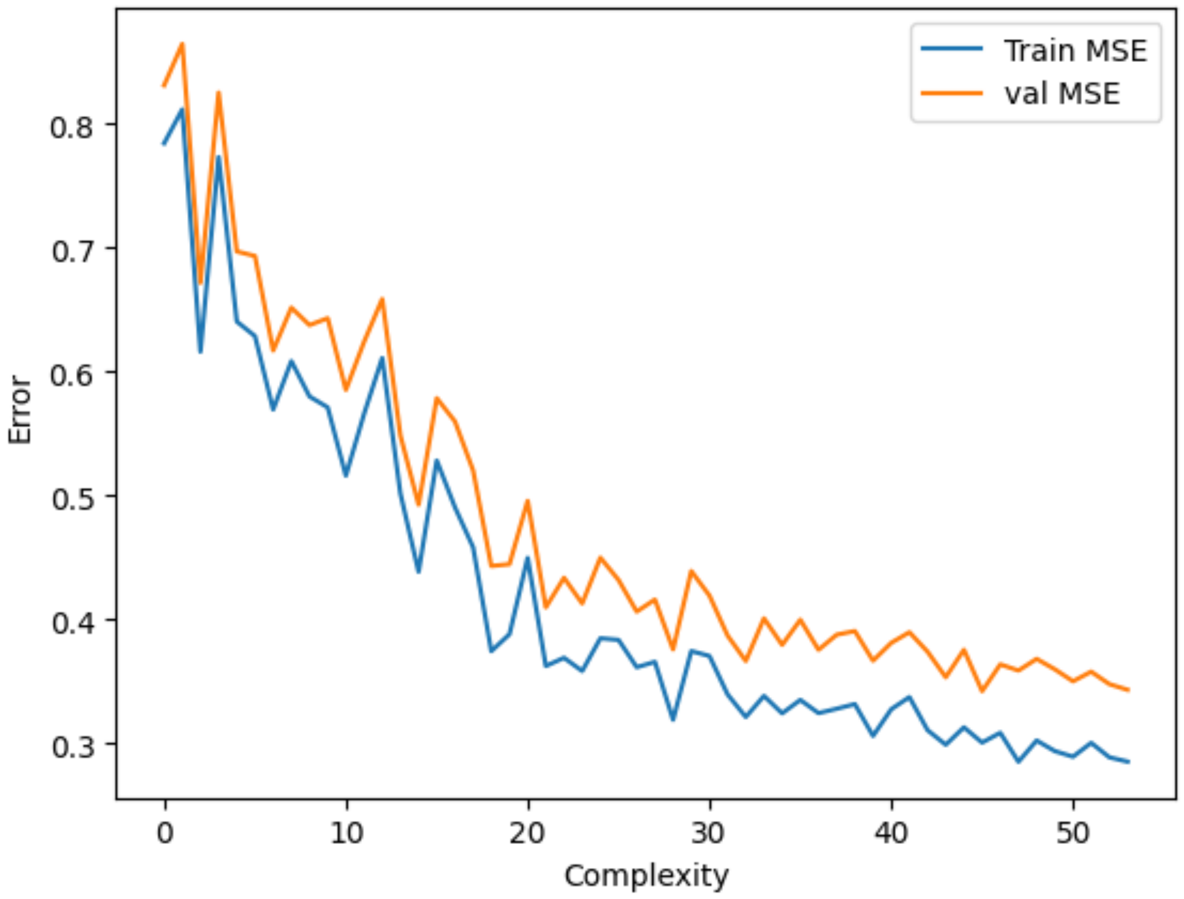
```
#Best Params for model
val_loss_list = []
train_loss_list = []

for i in range(6,41,2):
  model_complex_1 = create_ffnn(hidden_layers=[i+2,i,i],activation_functions=['
  history_complex_1 = train_model(model_complex_1,learning_rate=0.01,epochs=10(
  model_complex_2 = create_ffnn(hidden_layers=[i+2,i+2,i],activation_functions=
  history_complex_2 = train_model(model_complex_2,learning_rate=0.01,epochs=10(
  model_complex_3 = create_ffnn(hidden_layers=[i+2,i+2,i+2],activation_functior
  history_complex_3 = train_model(model_complex_3,learning_rate=0.01,epochs=10(
  train_loss_list.append(history_complex_1.history['loss'][-1])
  train_loss_list.append(history_complex_2.history['loss'][-1])
  train_loss_list.append(history_complex_3.history['loss'][-1])
  val_loss_list.append(history_complex_1.history['val_loss'][-1])
  val_loss_list.append(history_complex_2.history['val_loss'][-1])
  val_loss_list.append(history_complex_3.history['val_loss'][-1])
```

Görüntülenen çıkış son 5000 satıra kısaltıldı.

```
Epoch 1/100
32/32 - 2s - 56ms/step - loss: 2.4288 - mse: 2.4288 - val_loss: 2.0617 - va
Epoch 2/100
32/32 - 0s - 13ms/step - loss: 1.7754 - mse: 1.7754 - val_loss: 1.5901 - va
Epoch 3/100
32/32 - 0s - 5ms/step - loss: 1.4165 - mse: 1.4165 - val_loss: 1.3140 - val
Epoch 4/100
32/32 - 0s - 6ms/step - loss: 1.2000 - mse: 1.2000 - val_loss: 1.1412 - val
Epoch 5/100
32/32 - 0s - 9ms/step - loss: 1.0571 - mse: 1.0571 - val_loss: 1.0301 - val
Epoch 6/100
32/32 - 0s - 9ms/step - loss: 0.9643 - mse: 0.9643 - val_loss: 0.9633 - val
Epoch 7/100
32/32 - 0s - 5ms/step - loss: 0.9092 - mse: 0.9092 - val_loss: 0.9269 - val
Epoch 8/100
32/32 - 0s - 10ms/step - loss: 0.8782 - mse: 0.8782 - val_loss: 0.9061 - va
Epoch 9/100
32/32 - 0s - 9ms/step - loss: 0.8597 - mse: 0.8597 - val_loss: 0.8928 - val
Epoch 10/100
32/32 - 0s - 9ms/step - loss: 0.8468 - mse: 0.8468 - val_loss: 0.8831 - val
Epoch 11/100
32/32 - 0s - 9ms/step - loss: 0.8369 - mse: 0.8369 - val_loss: 0.8758 - val
```

```
Epoch 12/100
32/32 - 0s - 6ms/step - loss: 0.8286 - mse: 0.8286 - val_loss: 0.8693 - val
Epoch 13/100
32/32 - 0s - 6ms/step - loss: 0.8206 - mse: 0.8206 - val_loss: 0.8626 - val
Epoch 14/100
32/32 - 0s - 9ms/step - loss: 0.8134 - mse: 0.8134 - val_loss: 0.8566 - val
Epoch 15/100
32/32 - 0s - 6ms/step - loss: 0.8066 - mse: 0.8066 - val_loss: 0.8498 - val
Epoch 16/100
32/32 - 0s - 6ms/step - loss: 0.7996 - mse: 0.7996 - val_loss: 0.8449 - val
Epoch 17/100
32/32 - 0s - 9ms/step - loss: 0.7926 - mse: 0.7926 - val_loss: 0.8384 - val
Epoch 18/100
32/32 - 0s - 6ms/step - loss: 0.7860 - mse: 0.7860 - val_loss: 0.8319 - val
Epoch 19/100
32/32 - 0s - 6ms/step - loss: 0.7795 - mse: 0.7795 - val_loss: 0.8262 - val
Epoch 20/100
32/32 - 0s - 6ms/step - loss: 0.7727 - mse: 0.7727 - val_loss: 0.8202 - val
Epoch 21/100
32/32 - 0s - 9ms/step - loss: 0.7662 - mse: 0.7662 - val_loss: 0.8145 - val
Epoch 22/100
32/32 - 0s - 6ms/step - loss: 0.7592 - mse: 0.7592 - val_loss: 0.8078 - val
Epoch 23/100
32/32 - 0s - 6ms/step - loss: 0.7522 - mse: 0.7522 - val_loss: 0.8012 - val
Epoch 24/100
32/32 - 0s - 9ms/step - loss: 0.7441 - mse: 0.7441 - val_loss: 0.7949 - val
Epoch 25/100
32/32 - 0s - 9ms/step - loss: 0.7373 - mse: 0.7373 - val_loss: 0.7869 - val
Epoch 26/100
32/32 - 0s - 6ms/step - loss: 0.7299 - mse: 0.7299 - val_loss: 0.7797 - val
Epoch 27/100
32/32 - 0s - 9ms/step - loss: 0.7213 - mse: 0.7213 - val_loss: 0.7723 - val
Epoch 28/100
32/32 - 0s - 9ms/step - loss: 0.7129 - mse: 0.7129 - val_loss: 0.7636 - val
Epoch 29/100
32/32 - 0s - 9ms/step - loss: 0.7053 - mse: 0.7053 - val_loss: 0.7572 - val
```

```
plt.plot(train_loss_list, label='Train MSE')
plt.plot(val_loss_list, label='val MSE')
plt.xlabel('Complexity')
plt.ylabel('Error')
plt.legend()
plt.show()
```

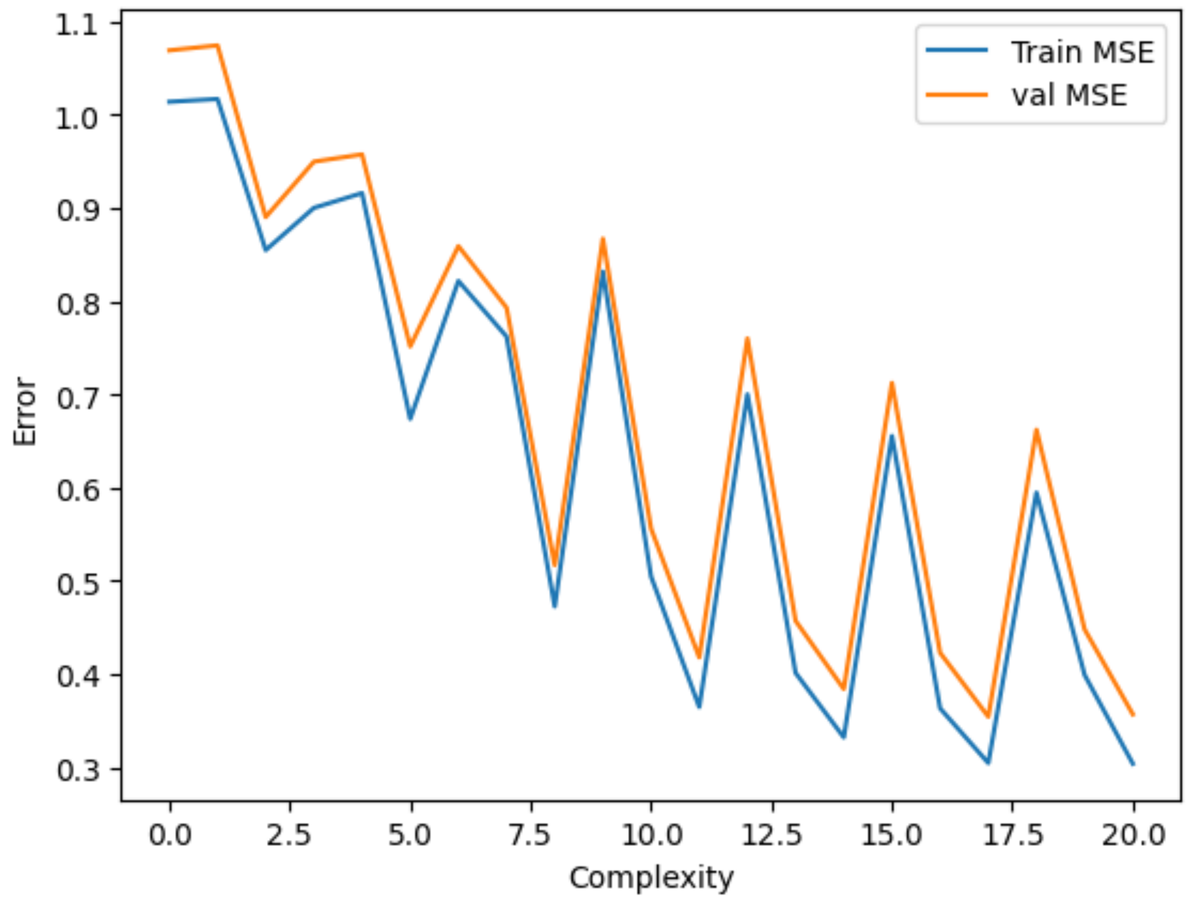


```
#Best Params for model
val_loss_list = []
train_loss_list = []

for i in range(1,8):
  model_complex_1 = create_ffnn(hidden_layers=[6*i,i,i],activation_functions=['
  history_complex_1 = train_model(model_complex_1,learning_rate=0.01,epochs=10(
  model_complex_2 = create_ffnn(hidden_layers=[6*i,6*i,i],activation_functions=
  history_complex_2 = train_model(model_complex_2,learning_rate=0.01,epochs=10(
  model_complex_3 = create_ffnn(hidden_layers=[6*i,6*i,6*i],activation_functior
  history_complex_3 = train_model(model_complex_3,learning_rate=0.01,epochs=10(
  train_loss_list.append(history_complex_1.history['loss'][-1])
  train_loss_list.append(history_complex_2.history['loss'][-1])
  train_loss_list.append(history_complex_3.history['loss'][-1])
  val_loss_list.append(history_complex_1.history['val_loss'][-1])
  val_loss_list.append(history_complex_2.history['val_loss'][-1])
  val_loss_list.append(history_complex_3.history['val_loss'][-1])
```

```
Epoch 1/100
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
32/32 - 1s - 31ms/step - loss: 2.5191 - mse: 2.5191 - val_loss: 2.3950 - va
Epoch 2/100
32/32 - 0s - 12ms/step - loss: 2.2312 - mse: 2.2312 - val_loss: 2.1289 - va
Epoch 3/100
32/32 - 0s - 9ms/step - loss: 2.0063 - mse: 2.0063 - val_loss: 1.9231 - val
Epoch 4/100
32/32 - 0s - 9ms/step - loss: 1.8337 - mse: 1.8337 - val_loss: 1.7603 - val
Epoch 5/100
32/32 - 0s - 10ms/step - loss: 1.6963 - mse: 1.6963 - val_loss: 1.6371 - va
Epoch 6/100
32/32 - 0s - 15ms/step - loss: 1.5896 - mse: 1.5896 - val_loss: 1.5322 - va
Epoch 7/100
32/32 - 0s - 6ms/step - loss: 1.5015 - mse: 1.5015 - val_loss: 1.4532 - val
Epoch 8/100
32/32 - 0s - 6ms/step - loss: 1.4322 - mse: 1.4322 - val_loss: 1.3959 - val
Epoch 9/100
32/32 - 0s - 10ms/step - loss: 1.3788 - mse: 1.3788 - val_loss: 1.3508 - va
Epoch 10/100
32/32 - 0s - 9ms/step - loss: 1.3377 - mse: 1.3377 - val_loss: 1.3142 - val
Epoch 11/100
32/32 - 0s - 7ms/step - loss: 1.3035 - mse: 1.3035 - val_loss: 1.2842 - val
Epoch 12/100
32/32 - 0s - 10ms/step - loss: 1.2739 - mse: 1.2739 - val_loss: 1.2595 - va
Epoch 13/100
32/32 - 0s - 6ms/step - loss: 1.2490 - mse: 1.2490 - val_loss: 1.2397 - val
Epoch 14/100
32/32 - 0s - 9ms/step - loss: 1.2282 - mse: 1.2282 - val_loss: 1.2232 - val
Epoch 15/100
32/32 - 0s - 6ms/step - loss: 1.2097 - mse: 1.2097 - val_loss: 1.2083 - val
Epoch 16/100
32/32 - 0s - 11ms/step - loss: 1.1933 - mse: 1.1933 - val_loss: 1.1948 - va
Epoch 17/100
32/32 - 0s - 12ms/step - loss: 1.1785 - mse: 1.1785 - val_loss: 1.1836 - va
Epoch 18/100
32/32 - 1s - 19ms/step - loss: 1.1651 - mse: 1.1651 - val_loss: 1.1743 - va
Epoch 19/100
32/32 - 1s - 20ms/step - loss: 1.1536 - mse: 1.1536 - val_loss: 1.1656 - va
Epoch 20/100
32/32 - 0s - 6ms/step - loss: 1.1426 - mse: 1.1426 - val_loss: 1.1571 - val
Epoch 21/100
32/32 - 0s - 9ms/step - loss: 1.1330 - mse: 1.1330 - val_loss: 1.1497 - val
Epoch 22/100
32/32 - 0s - 10ms/step - loss: 1.1241 - mse: 1.1241 - val_loss: 1.1427 - va
Epoch 23/100
32/32 - 0s - 9ms/step - loss: 1.1158 - mse: 1.1158 - val_loss: 1.1373 - val
```

```
Epoch 24/100
32/32 - 0s - 8ms/step - loss: 1.1085 - mse: 1.1085 - val_loss: 1.1312 - val
Epoch 25/100
32/32 - 0s - 6ms/step - loss: 1.1017 - mse: 1.1017 - val_loss: 1.1261 - val
Epoch 26/100
32/32 - 0s - 5ms/step - loss: 1.0956 - mse: 1.0956 - val_loss: 1.1211 - val
Epoch 27/100
32/32 - 0s - 5ms/step - loss: 1.0894 - mse: 1.0894 - val_loss: 1.1171 - val
Epoch 28/100
32/32 - 0s - 10ms/step - loss: 1.0843 - mse: 1.0843 - val_loss: 1.1134 - va
Epoch 29/100
```

```
plt.plot(train_loss_list, label='Train MSE')
plt.plot(val_loss_list, label='val MSE')
plt.xlabel('Complexity')
plt.ylabel('Error')
plt.legend()
plt.show()
```



```
# Calculate the increased training data size
increased_size = int(N_t * 0.1)
new_N_t = N_t + increased_size

# Generate additional training data
X_train_add, Y_train_add = generate_data(increased_size, seed=101) # Use a dif

# Concatenate the original and additional data
X_train_increased = np.concatenate((X_train, X_train_add))
Y_train_increased = np.concatenate((Y_train, Y_train_add))

# Verify new training data size
print(f"Original training data size: {N_t}")
print(f"Increased training data size: {increased_size}")
print(f"New training data size: {len(X_train_increased)}")
```

```
Original training data size: 1000
Increased training data size: 100
New training data size: 1100
```

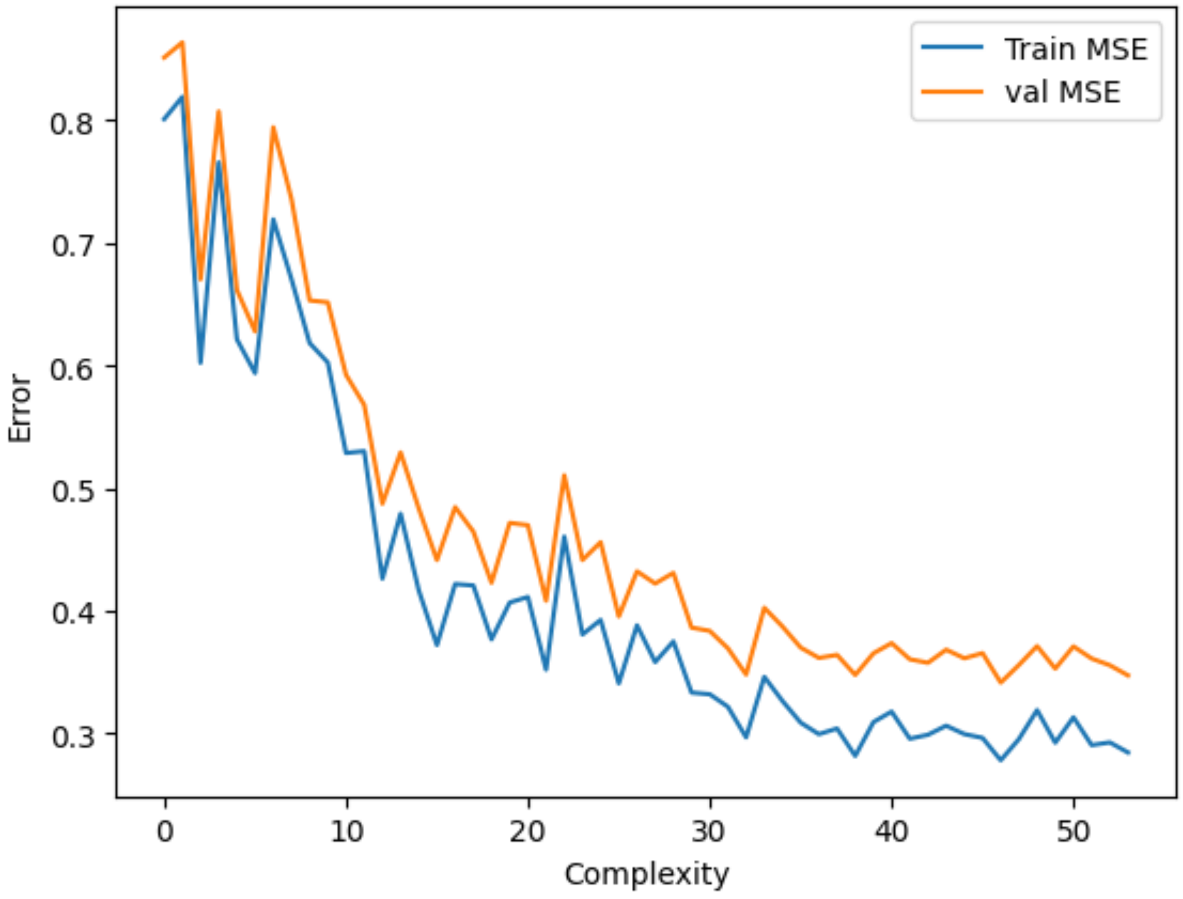
```
#Best Params for model
val_loss_list = []
train_loss_list = []

for i in range(6,41,2):
  model_complex_1 = create_ffnn(hidden_layers=[i+2,i,i],activation_functions=['
  history_complex_1 = train_model(model_complex_1,learning_rate=0.01,epochs=10
  model_complex_2 = create_ffnn(hidden_layers=[i+2,i+2,i],activation_functions=
  history_complex_2 = train_model(model_complex_2,learning_rate=0.01,epochs=10
  model_complex_3 = create_ffnn(hidden_layers=[i+2,i+2,i+2],activation_functio
  history_complex_3 = train_model(model_complex_3,learning_rate=0.01,epochs=10
  train_loss_list.append(history_complex_1.history['loss'][-1])
  train_loss_list.append(history_complex_2.history['loss'][-1])
  train_loss_list.append(history_complex_3.history['loss'][-1])
  val_loss_list.append(history_complex_1.history['val_loss'][-1])
  val_loss_list.append(history_complex_2.history['val_loss'][-1])
  val_loss_list.append(history_complex_3.history['val_loss'][-1])
```

```
Epoch 1/100
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```
Görüntülenen çıkış son 5000 satıra kısaltıldı.
```
Epoch 1/100
```

```
35/35 - 1s - 32ms/step - loss: 2.3142 - mse: 2.3142 - val_loss: 1.8346 - va
Epoch 2/100
35/35 - 0s - 14ms/step - loss: 1.4912 - mse: 1.4912 - val_loss: 1.3354 - va
Epoch 3/100
35/35 - 0s - 10ms/step - loss: 1.1845 - mse: 1.1845 - val_loss: 1.1545 - va
Epoch 4/100
35/35 - 0s - 5ms/step - loss: 1.0696 - mse: 1.0696 - val_loss: 1.0744 - val
Epoch 5/100
35/35 - 0s - 5ms/step - loss: 1.0096 - mse: 1.0096 - val_loss: 1.0279 - val
Epoch 6/100
35/35 - 0s - 9ms/step - loss: 0.9708 - mse: 0.9708 - val_loss: 0.9970 - val
Epoch 7/100
35/35 - 0s - 5ms/step - loss: 0.9432 - mse: 0.9432 - val_loss: 0.9744 - val
Epoch 8/100
35/35 - 0s - 8ms/step - loss: 0.9220 - mse: 0.9220 - val_loss: 0.9560 - val
Epoch 9/100
35/35 - 0s - 9ms/step - loss: 0.9040 - mse: 0.9040 - val_loss: 0.9389 - val
Epoch 10/100
35/35 - 0s - 6ms/step - loss: 0.8875 - mse: 0.8875 - val_loss: 0.9243 - val
Epoch 11/100
35/35 - 0s - 5ms/step - loss: 0.8721 - mse: 0.8721 - val_loss: 0.9096 - val
Epoch 12/100
35/35 - 0s - 9ms/step - loss: 0.8575 - mse: 0.8575 - val_loss: 0.8956 - val
Epoch 13/100
35/35 - 0s - 8ms/step - loss: 0.8439 - mse: 0.8439 - val_loss: 0.8828 - val
Epoch 14/100
35/35 - 0s - 5ms/step - loss: 0.8316 - mse: 0.8316 - val_loss: 0.8703 - val
Epoch 15/100
35/35 - 0s - 9ms/step - loss: 0.8199 - mse: 0.8199 - val_loss: 0.8580 - val
Epoch 16/100
35/35 - 0s - 5ms/step - loss: 0.8086 - mse: 0.8086 - val_loss: 0.8464 - val
Epoch 17/100
35/35 - 0s - 9ms/step - loss: 0.7978 - mse: 0.7978 - val_loss: 0.8358 - val
Epoch 18/100
35/35 - 0s - 9ms/step - loss: 0.7882 - mse: 0.7882 - val_loss: 0.8248 - val
Epoch 19/100
35/35 - 0s - 5ms/step - loss: 0.7787 - mse: 0.7787 - val_loss: 0.8145 - val
Epoch 20/100
35/35 - 0s - 9ms/step - loss: 0.7694 - mse: 0.7694 - val_loss: 0.8053 - val
Epoch 21/100
35/35 - 0s - 6ms/step - loss: 0.7606 - mse: 0.7606 - val_loss: 0.7954 - val
Epoch 22/100
35/35 - 0s - 8ms/step - loss: 0.7521 - mse: 0.7521 - val_loss: 0.7860 - val
Epoch 23/100
35/35 - 0s - 9ms/step - loss: 0.7440 - mse: 0.7440 - val_loss: 0.7777 - val
Epoch 24/100
35/35 - 0s - 9ms/step - loss: 0.7359 - mse: 0.7359 - val_loss: 0.7693 - val
Epoch 25/100
35/35 - 0s - 8ms/step - loss: 0.7274 - mse: 0.7274 - val_loss: 0.7616 - val
Epoch 26/100
35/35 - 0s - 9ms/step - loss: 0.7194 - mse: 0.7194 - val_loss: 0.7520 - val
Epoch 27/100
35/35 - 0s - 5ms/step - loss: 0.7110 - mse: 0.7110 - val_loss: 0.7446 - val
```

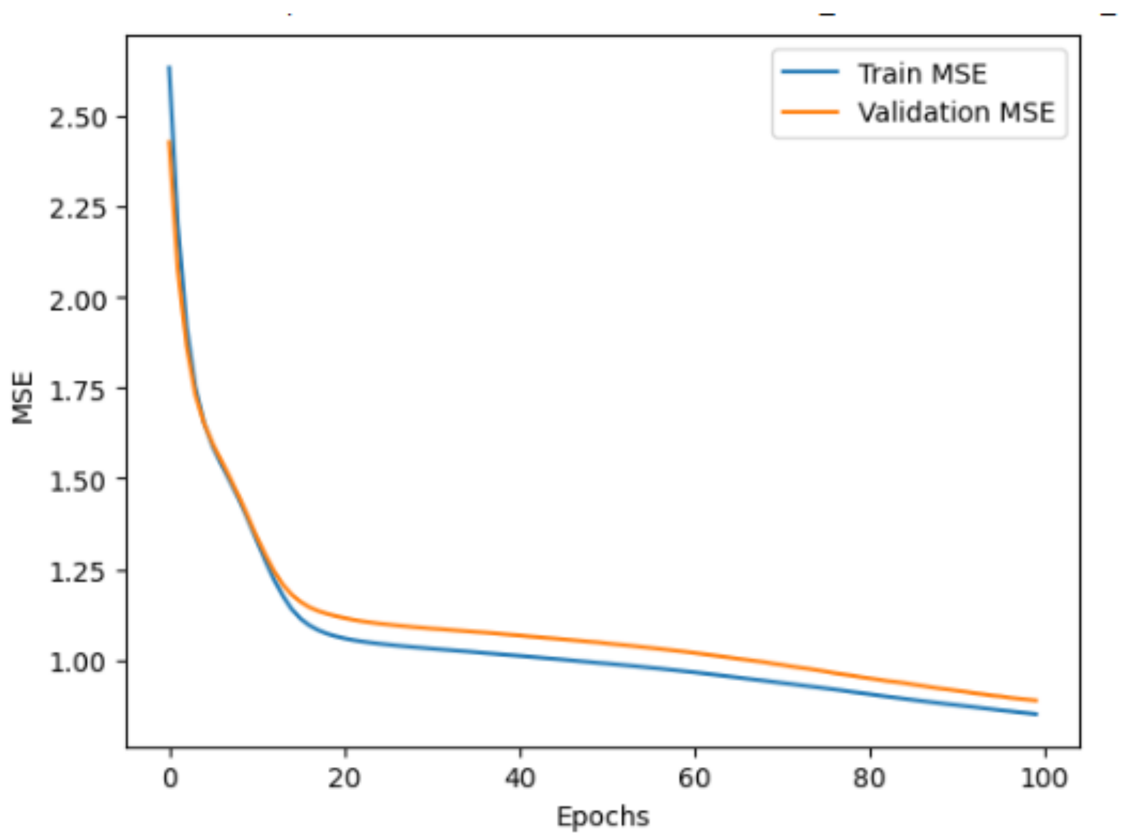
```
Epoch 28/100
```

```
plt.plot(train_loss_list, label='Train MSE')
plt.plot(val_loss_list, label='val MSE')
plt.xlabel('Complexity')
plt.ylabel('Error')
plt.legend()
plt.show()
```



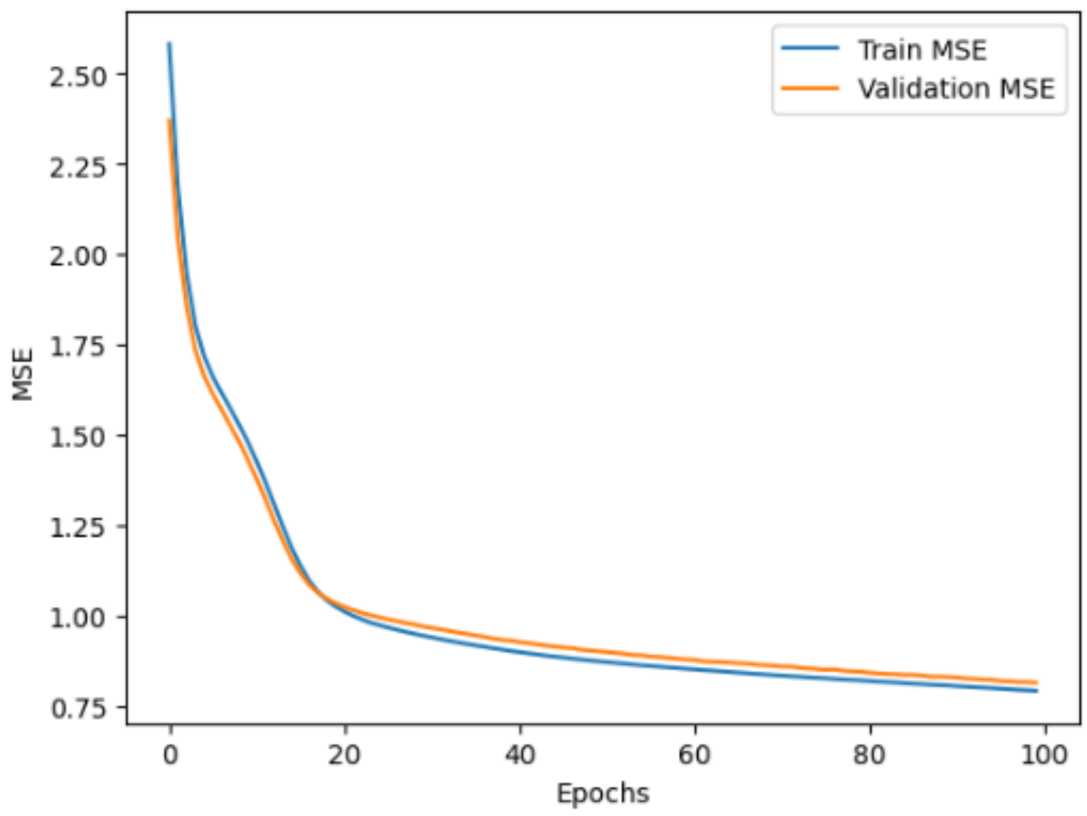
Results:

Step 5 : First Model Train Loss Validation over Epoch Graph with given params

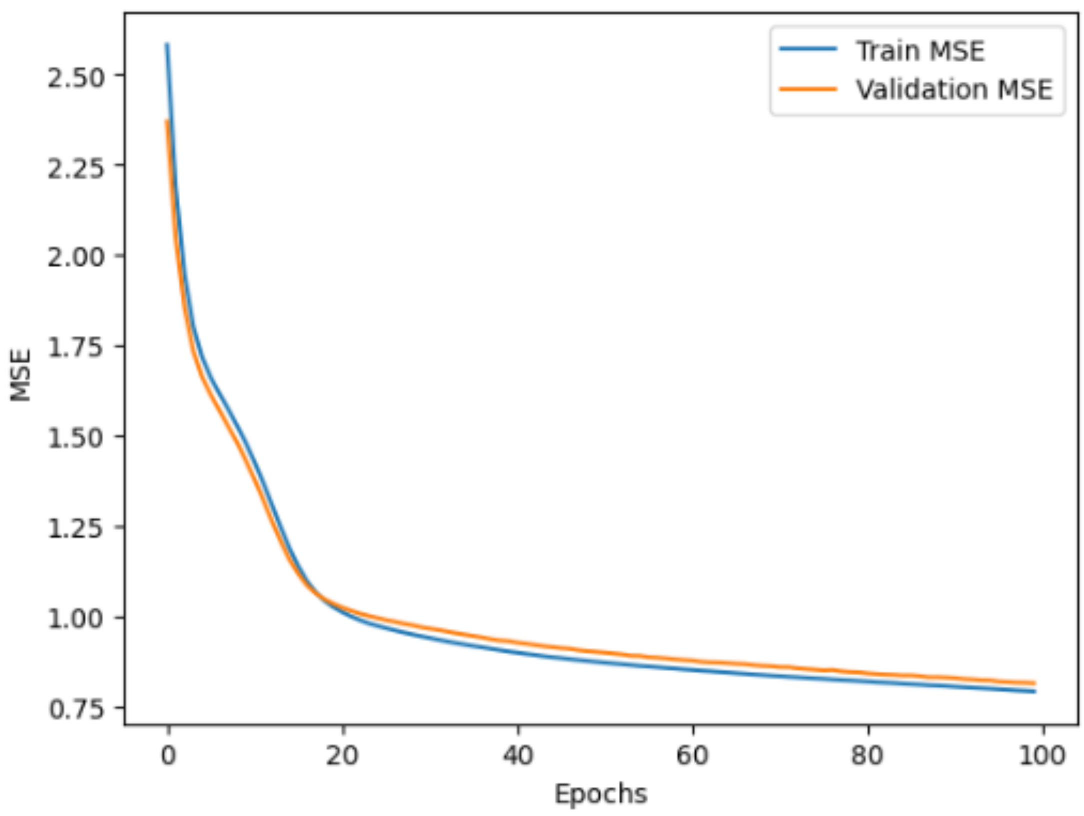


Step 6: Train Validation Loss of Deep Learning Model over Epoch with Different Set of Activation Functions

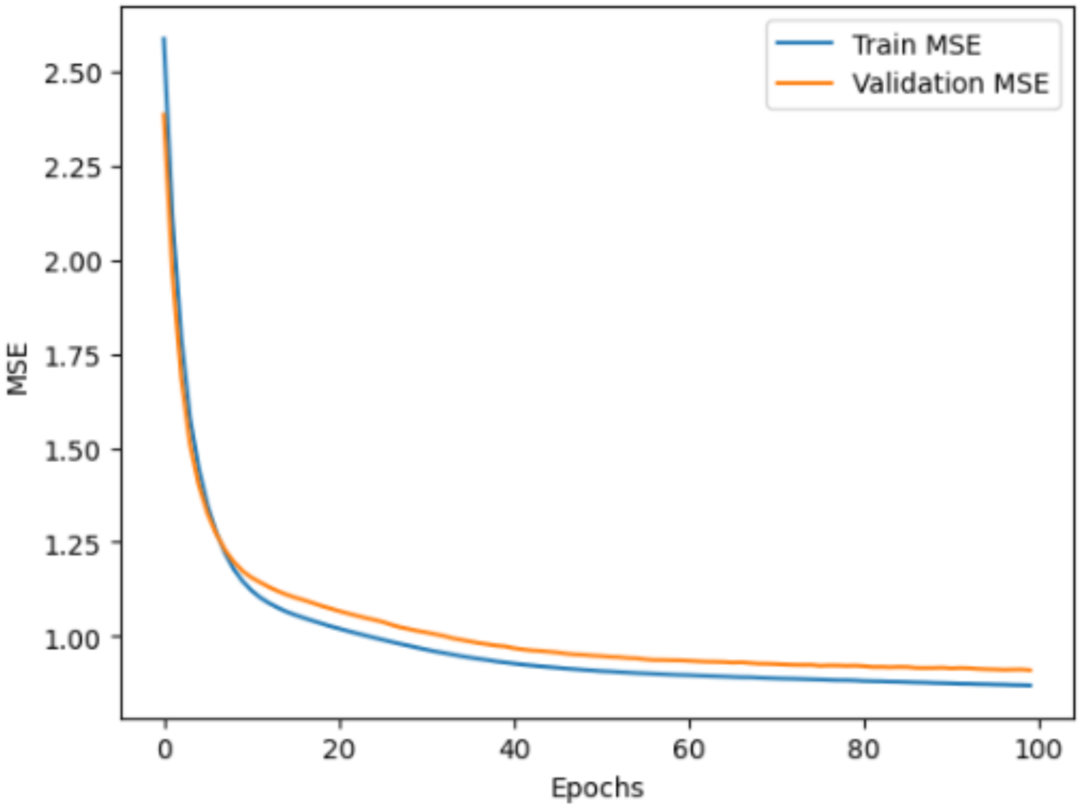
activation_functions=['tanh', 'leaky_relu', 'linear']



activation_functions=['relu', 'relu', 'linear']

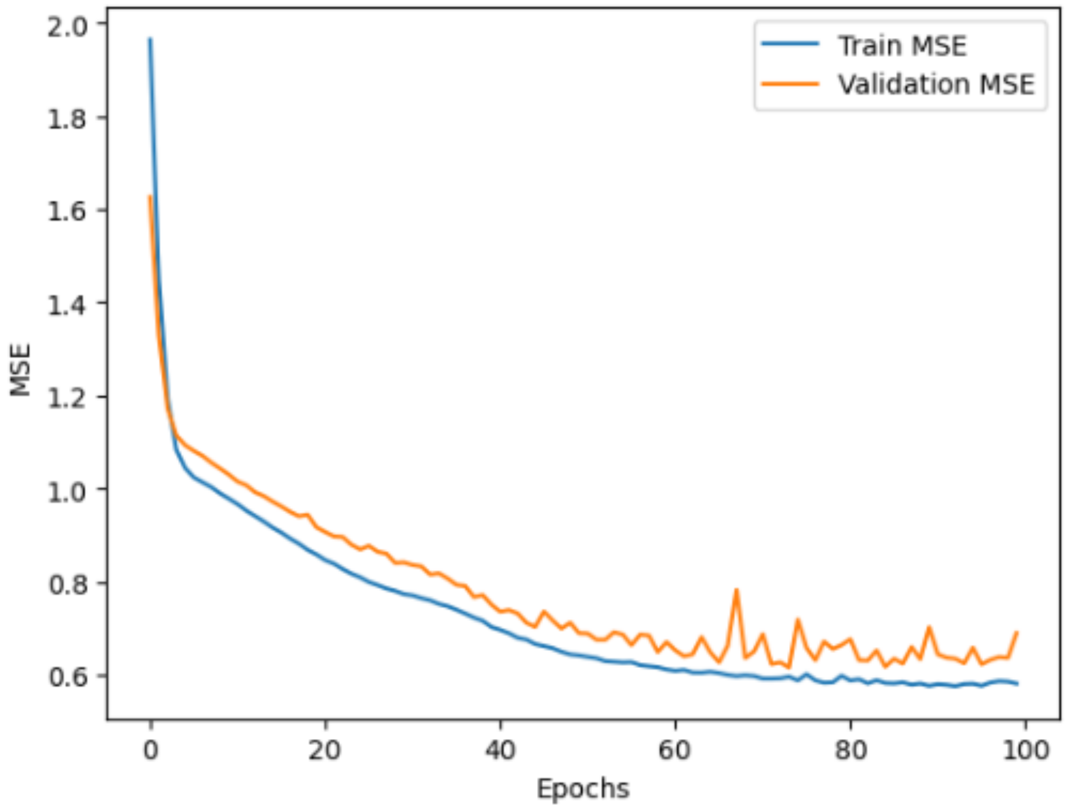


activation_functions=['tanh', 'leaky_relu', 'leaky_relu']

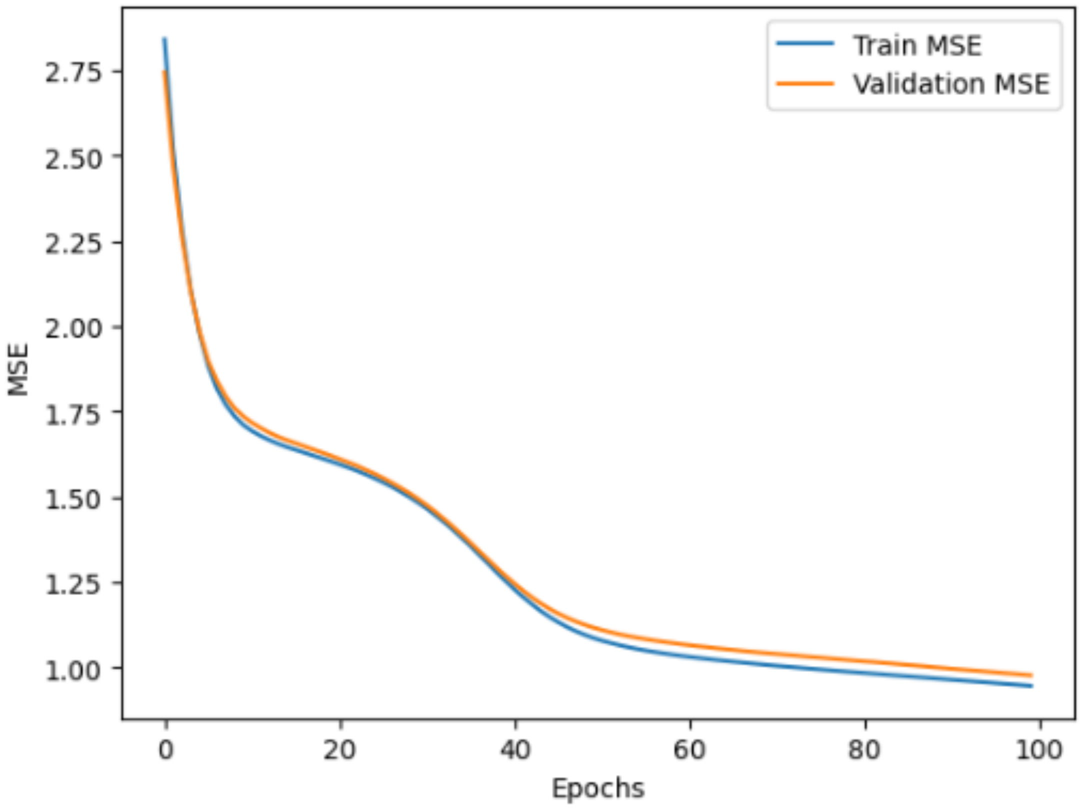


Step 6: Train Validation Loss of Model over Epoch with Different Learning Rates :

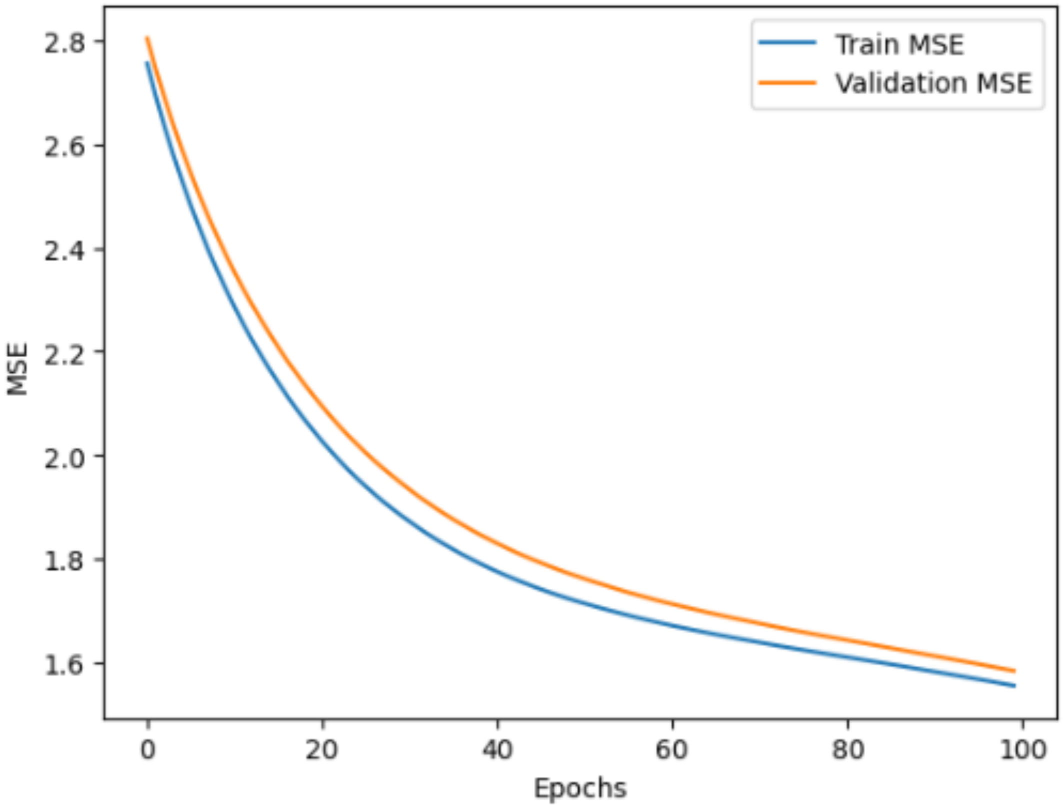
Learning Rate = 0.05



Learning Rate = 0.005

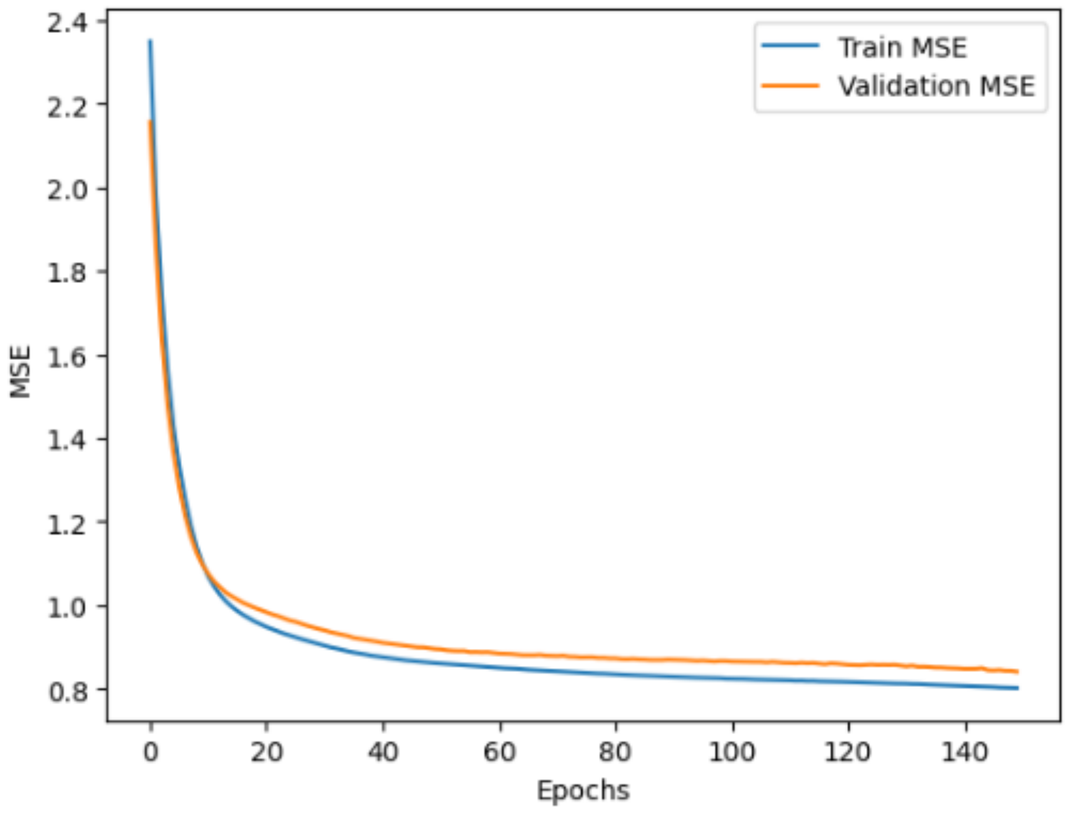


Learning Rate = 0.001

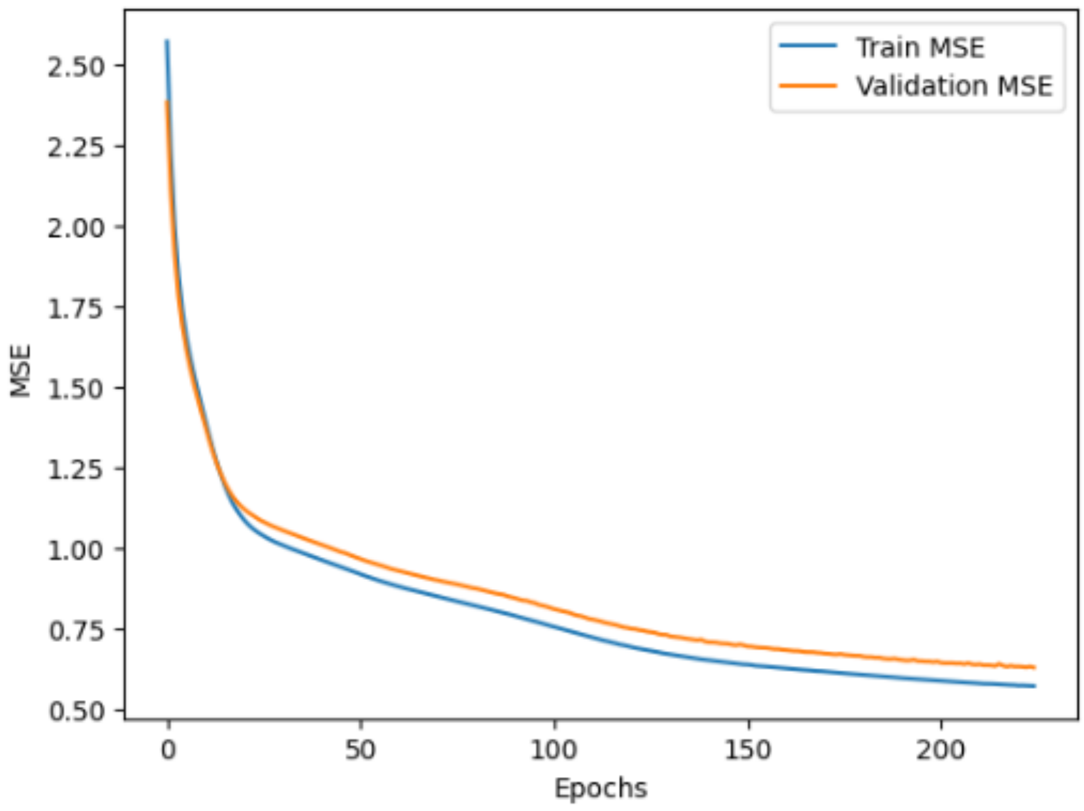


Step 6: Train Validation Loss of Model over Epoch with Different Epochs :

Epoch increased %50 = 150 Epoch

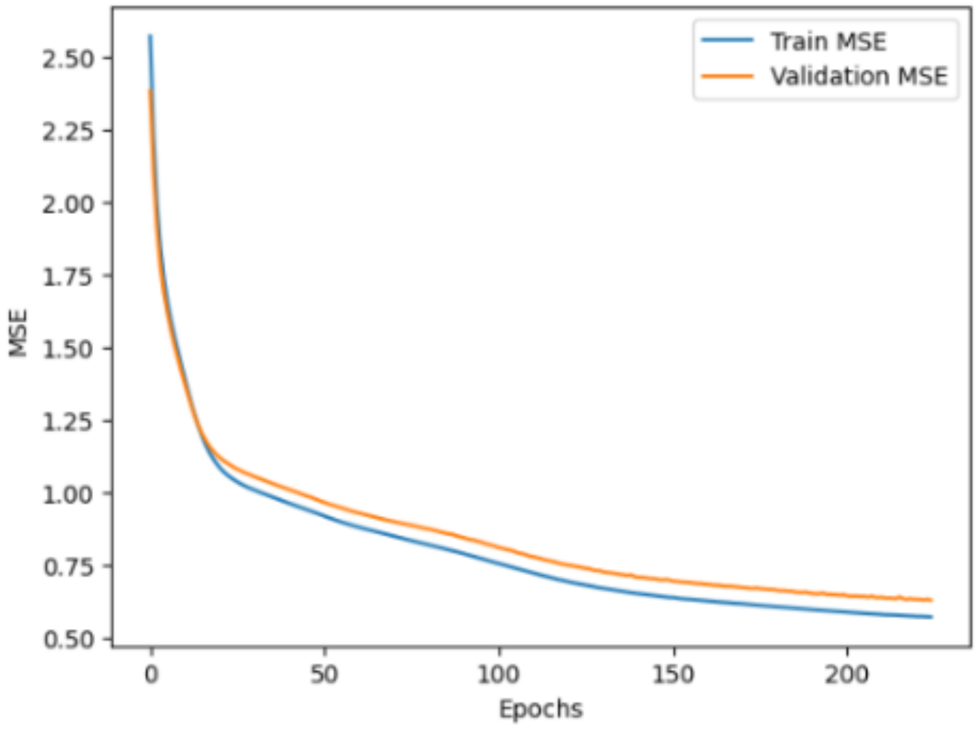


Epoch increased %50 more = 225 Epoch



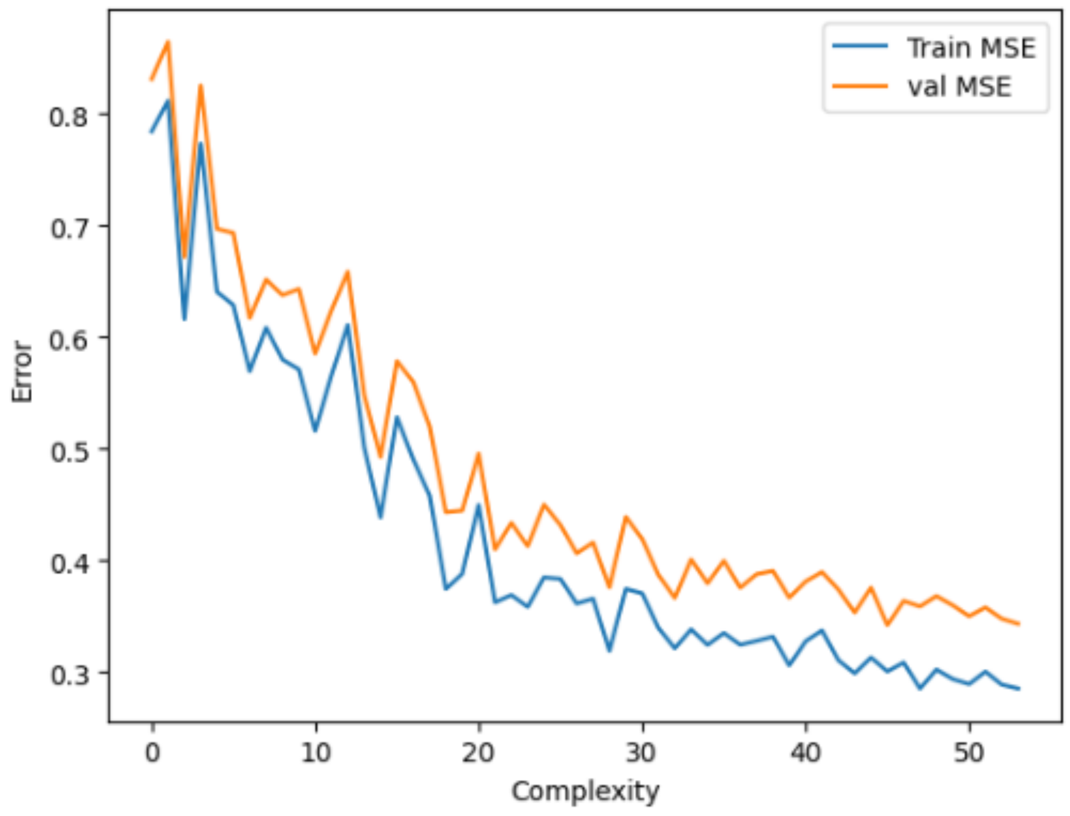
Step 7: Train Validation Loss of Model with Best Params

- Epoch = 225
- Learning Rate = 0.01
- Activation Functions = ['tanh', 'leaky_relu', 'linear']

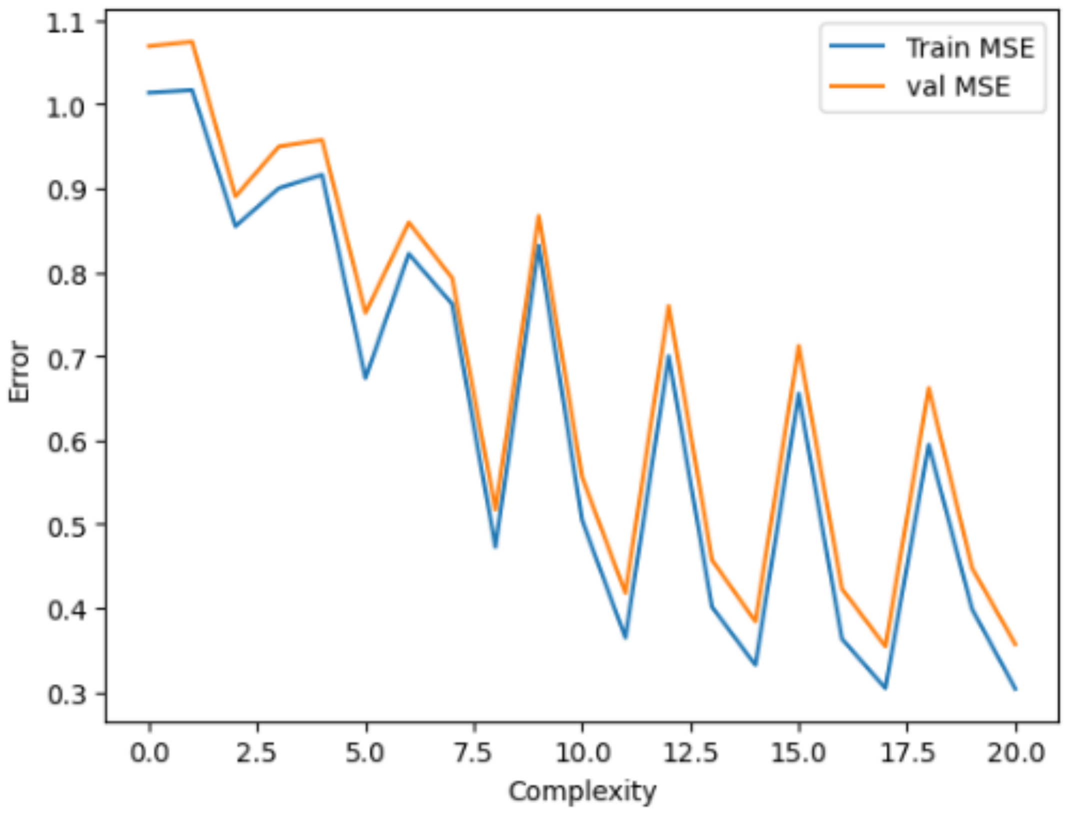


Step 8: Add new nodes at a time to each hidden layer :

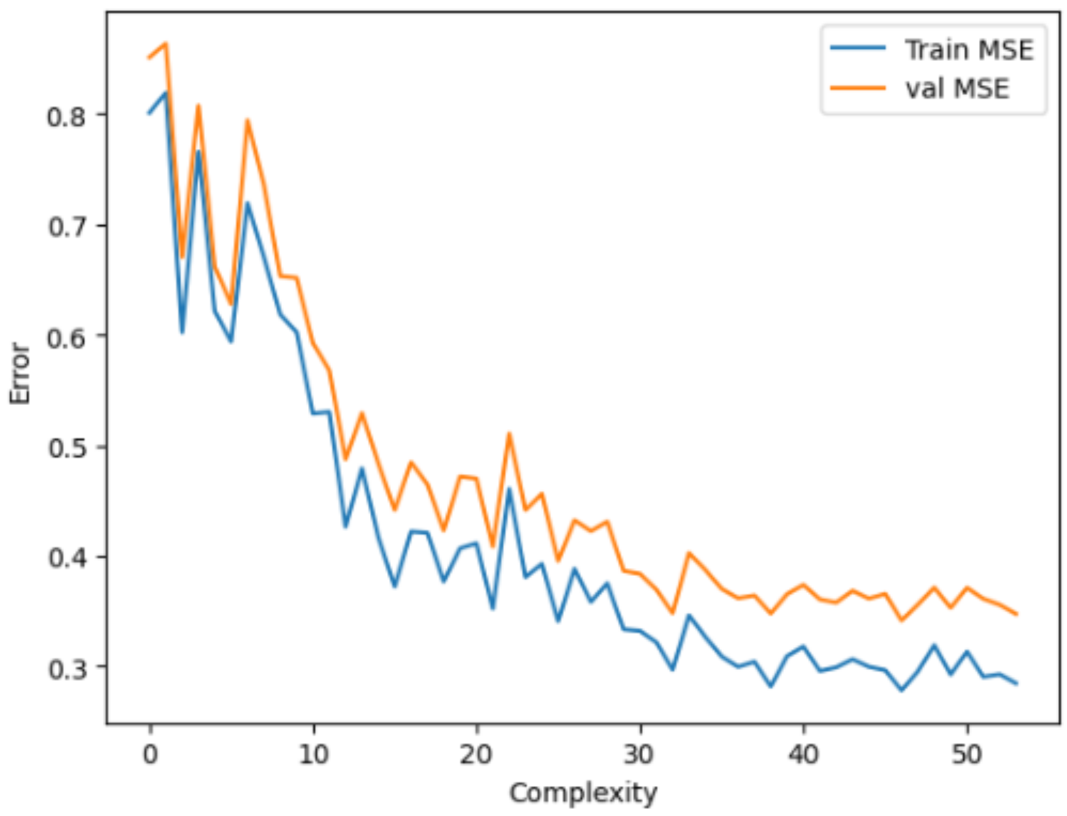
This process continue until each hidden layer reach the 80 nodes



Then I tried to double node counts each time like 12,24,48,96,192,... It reached 720 nodes for each layer



Step 9 : Increased Training Data as %10 and Repeat Step 8



Comments and Discussions :

- 1000 data pairs(-1 to 1) for training and 500 data pairs for validation generated using given polynomal equations.
- Noise added to %30 of 1000 data training data.
- There is a seed value for this part to generate same data during process.
- Increased noisy data up to a limit in dl model cause reduced to overfit and increasing the robustness of model. Therefore , its an important part.
- Deep Learning Model created using Keras library Sequantial to create Feed Forward Neural Network. Each hidden layer contains 6 nodes and same activation function in same layer. Also 8 input and 5 output contains.
- MSE is used for loss function and model trained using SGD.

Step 5 :

1. Activation Functions = ['relu', 'tanh', 'tanh']

2. learning_rate=0.01
3. epochs=100

- Actually all of the params choosed randomly checking some interval of numbers and when look at the model , everyway it looks not good because mse is 0.8 for data between(-1,1). It was high value for this case probably but can not find better optimal solution . It could be about hidden layers , some layer changes could be solution because after step 8 i realized it reduce when complexity increase (bias-variance problem) but layer changes except except node counts is not case of this project so i continued.

Step 6:

- I realized by trying different combinations that the sigmoid function is not good for this problem and that being linear in the last layer and having a leaky relu in one of the layers gives better results in general.
- I also tried different learning rates and concluded that some learning rates (0.001) do not learn enough and are underfit, some (0.05) try to learn fast and do not learn properly and miss the global or local minima of the loss function, and I decided that 0.01 is the optimal learning rate.
- With the optimal learning rate and activation functions I chose, I noticed that it learned more and more as I increased the epoch amount by 50%, and I concluded that it learned by progressing optimally for the last 225 epochs I tried.

Step 7:

- I choosed these params for my problem solution. These params given best results i have tried before. It learns optimal and not overfitting or underfitting also learning properly.Error was still high but i continued.
- Epoch = 225
- Learning Rate = 0.01
- Activation Functions = ['tanh', 'leaky_relu', 'linear']

Step 8:

- First I increased the number of nodes in each layer by two and continued this process until I had 80 nodes in each layer, but I could not get the optimal bias-variance trade-off graph. Also, since this model is time consuming, I used epoch 100. Despite these values, the process took longer. Afterwards, I wanted to do a different experiment by taking two multiples (6,12,24,48,96,192) of the nodes in each layer continuously. I made 768 nodes in each layer this way, but I may not have achieved an optimal graph because I am constantly doubling it. Again, there was a graph that was constantly going down, but at some points it can be likened to the increase in errors when going to the high variance low bias part.

Step 9:

- I continued by adding two nodes to each layer as I did in the first part in step 8, but I don't think there is a noticeable difference in the graph. However, I think that maybe the situation that should be here is that the error value decreases by moving both validation and training closer to the bottom of the graph, or that the gap between train and validation closes a little more and the model learns better.

Part 2:

```
dataset_path = "/content/drive/MyDrive/Shapes/"
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator


# Define ImageDataGenerator with rescaling
datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)  # 80% trair

# Load Training Data
train_generator = datagen.flow_from_directory(
    dataset_path + "Train/",
    target_size=(128, 128),
    color_mode="grayscale",  # Since we have grayscale images
    batch_size=32,
    class_mode="categorical",
    subset="training"
)

# Load Validation Data
val_generator = datagen.flow_from_directory(
    dataset_path + "Train/",
    target_size=(128, 128),
    color_mode="grayscale",
    batch_size=32,
    class_mode="categorical",
    subset="validation"
)
```

```
Found 1319 images belonging to 7 classes.
Found 328 images belonging to 7 classes.
```

```
import tensorflow as tf
from tensorflow.keras import layers, models

def modified_alexnet(input_shape=(128, 128, 1), num_classes=7):
    model = models.Sequential()

    # First Convolutional Layer (Modified for 1-Channel Input)
    model.add(layers.Conv2D(96, kernel_size=(11, 11), strides=(4, 4), activatic
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Second Convolutional Layer
    model.add(layers.Conv2D(256, kernel_size=(5, 5), activation='relu', padding
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Third, Fourth, and Fifth Convolutional Layers
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(256, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Flattening
    model.add(layers.Flatten())

    # Fully Connected Layers
    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))
    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))

    # Output Layer (Change to 8 classes for shape classification)
    model.add(layers.Dense(num_classes, activation='softmax'))

    return model

# Create model
model = modified_alexnet()
model.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Model: "sequential"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━
┃ Layer (type)                         ┃ Output Shape                ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━
│ conv2d (Conv2D)                      │ (None, 30, 30, 96)          │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ batch_normalization                  │ (None, 30, 30, 96)          │
│ (BatchNormalization)                 │                             │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ max_pooling2d (MaxPooling2D)         │ (None, 14, 14, 96)          │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ conv2d_1 (Conv2D)                    │ (None, 14, 14, 256)         │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ batch_normalization_1                │ (None, 14, 14, 256)         │
│ (BatchNormalization)                 │                             │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ max_pooling2d_1 (MaxPooling2D)       │ (None, 6, 6, 256)           │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ conv2d_2 (Conv2D)                    │ (None, 6, 6, 384)           │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ conv2d_3 (Conv2D)                    │ (None, 6, 6, 384)           │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ conv2d_4 (Conv2D)                    │ (None, 6, 6, 256)           │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ max_pooling2d_2 (MaxPooling2D)       │ (None, 2, 2, 256)           │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ flatten (Flatten)                    │ (None, 1024)                │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ dense (Dense)                        │ (None, 4096)                │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ dropout (Dropout)                    │ (None, 4096)                │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ dense_1 (Dense)                      │ (None, 4096)                │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ dropout_1 (Dropout)                  │ (None, 4096)                │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ dense_2 (Dense)                      │ (None, 7)                   │
└──────────────────────────────────────┴─────────────────────────────┴─────
 Total params: 24,733,767 (94.35 MB)
 Trainable params: 24,733,063 (94.35 MB)
 Non-trainable params: 704 (2.75 KB)
```

```
model.fit(train_generator, validation_data=val_generator, epochs=4, batch_size=
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py
  self._warn_if_super_not_called()
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 348s 8s/step - accuracy: 0.1565 - loss: 2.2073 -
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 81s 2s/step - accuracy: 0.1583 - loss: 1.9394 -
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 83s 2s/step - accuracy: 0.1867 - loss: 1.9155 -
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 139s 2s/step - accuracy: 0.1878 - loss: 1.9038 -
<keras.src.callbacks.history.History at 0x7e70e38ac1d0>
```

```
import tensorflow as tf
from tensorflow.keras import layers, models

def modified_alexnet(input_shape=(128, 128, 1), num_classes=7):
    model = models.Sequential()

    # First Convolutional Layer (Modified for 1-Channel Input)
    model.add(layers.Conv2D(96, kernel_size=(11, 11), strides=(4, 4), activatic
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Second Convolutional Layer
    model.add(layers.Conv2D(256, kernel_size=(5, 5), activation='relu', padding
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Third, Fourth, and Fifth Convolutional Layers
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(256, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Flattening
    model.add(layers.Flatten())

    # Fully Connected Layers
    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))
    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))

    # Output Layer (Change to 8 classes for shape classification)
    model.add(layers.Dense(num_classes, activation='softmax'))
    return model

# Create model
model = modified_alexnet()
model.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.001, momentum=(
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
Model: "sequential_1"
```

Layer (type)	Output Shape
conv2d_5 (Conv2D)	(None, 30, 30, 96)
batch_normalization_2 (BatchNormalization)	(None, 30, 30, 96)
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 96)
conv2d_6 (Conv2D)	(None, 14, 14, 256)
batch_normalization_3 (BatchNormalization)	(None, 14, 14, 256)
max_pooling2d_4 (MaxPooling2D)	(None, 6, 6, 256)
conv2d_7 (Conv2D)	(None, 6, 6, 384)
conv2d_8 (Conv2D)	(None, 6, 6, 384)
conv2d_9 (Conv2D)	(None, 6, 6, 256)
max_pooling2d_5 (MaxPooling2D)	(None, 2, 2, 256)
flatten_1 (Flatten)	(None, 1024)
dense_3 (Dense)	(None, 4096)
dropout_2 (Dropout)	(None, 4096)
dense_4 (Dense)	(None, 4096)
dropout_3 (Dropout)	(None, 4096)
dense_5 (Dense)	(None, 7)

```
Total params: 24,733,767 (94.35 MB)
Trainable params: 24,733,063 (94.35 MB)
Non-trainable params: 704 (2.75 KB)
```

```
model.fit(train_generator, validation_data=val_generator, epochs=4, batch_size=
```

```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 87s 2s/step - accuracy: 0.1259 - loss: 2.2454 -
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.1500 - loss: 1.9725 -
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 84s 2s/step - accuracy: 0.1813 - loss: 1.9409 -
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 75s 2s/step - accuracy: 0.1924 - loss: 1.9130 -
<keras.src.callbacks.history.History at 0x7e70d41f41d0>
```

```
import tensorflow as tf
from tensorflow.keras import layers, models

def modified_alexnet(input_shape=(128, 128, 1), num_classes=7):
    model = models.Sequential()

    # First Convolutional Layer (Modified for 1-Channel Input)
    model.add(layers.Conv2D(96, kernel_size=(11, 11), strides=(4, 4), activatic
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Second Convolutional Layer
    model.add(layers.Conv2D(256, kernel_size=(5, 5), activation='relu', paddinc
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Third, Fourth, and Fifth Convolutional Layers
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='tanh', paddinc
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', paddinc
    model.add(layers.Conv2D(256, kernel_size=(3, 3), activation='tanh', paddinc
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Flattening
    model.add(layers.Flatten())

    # Fully Connected Layers
    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))
    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))

    # Output Layer (Change to 8 classes for shape classification)
    model.add(layers.Dense(num_classes, activation='softmax'))

    return model
```

```
# Create model
model = modified_alexnet()
model.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.001, momentum=(
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
Model: "sequential_2"
```

Layer (type)	Output Shape	
conv2d_10 (Conv2D)	(None, 30, 30, 96)	
batch_normalization_4 (BatchNormalization)	(None, 30, 30, 96)	
max_pooling2d_6 (MaxPooling2D)	(None, 14, 14, 96)	
conv2d_11 (Conv2D)	(None, 14, 14, 256)	
batch_normalization_5 (BatchNormalization)	(None, 14, 14, 256)	
max_pooling2d_7 (MaxPooling2D)	(None, 6, 6, 256)	
conv2d_12 (Conv2D)	(None, 6, 6, 384)	
conv2d_13 (Conv2D)	(None, 6, 6, 384)	
conv2d_14 (Conv2D)	(None, 6, 6, 256)	
max_pooling2d_8 (MaxPooling2D)	(None, 2, 2, 256)	
flatten_2 (Flatten)	(None, 1024)	
dense_6 (Dense)	(None, 4096)	
dropout_4 (Dropout)	(None, 4096)	
dense_7 (Dense)	(None, 4096)	
dropout_5 (Dropout)	(None, 4096)	
dense_8 (Dense)	(None, 7)	

```
Total params: 24,733,767 (94.35 MB)
Trainable params: 24,733,063 (94.35 MB)
Non-trainable params: 704 (2.75 KB)
```

```
model.fit(train_generator, validation_data=val_generator, epochs=4, batch_size=
```

```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 80s 2s/step - accuracy: 0.1290 - loss: 2.0077 -
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 89s 2s/step - accuracy: 0.1584 - loss: 1.9464 -
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 85s 2s/step - accuracy: 0.1549 - loss: 1.9441 -
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 142s 2s/step - accuracy: 0.1925 - loss: 1.9115 -
<keras.src.callbacks.history.History at 0x7e70d42bc1d0>
```

```
import tensorflow as tf
from tensorflow.keras import layers, models

def modified_alexnet(input_shape=(128, 128, 1), num_classes=7):
    model = models.Sequential()

    # First Convolutional Layer (Modified for 1-Channel Input)
    model.add(layers.Conv2D(96, kernel_size=(11, 11), strides=(4, 4), activatic
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Second Convolutional Layer
    model.add(layers.Conv2D(256, kernel_size=(5, 5), activation='sigmoid', padc
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Third, Fourth, and Fifth Convolutional Layers
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='leaky_relu', p
    model.add(layers.Conv2D(256, kernel_size=(3, 3), activation='tanh', padding
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Flattening
    model.add(layers.Flatten())

    # Fully Connected Layers
    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))
    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))

    # Output Layer (Change to 8 classes for shape classification)
    model.add(layers.Dense(num_classes, activation='softmax'))

    return model
```

```
# Create model
model = modified_alexnet()
model.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.001, momentum=0
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Model: "sequential_3"
```

Layer (type)	Output Shape	
conv2d_15 (Conv2D)	(None, 30, 30, 96)	
batch_normalization_6 (BatchNormalization)	(None, 30, 30, 96)	
max_pooling2d_9 (MaxPooling2D)	(None, 14, 14, 96)	
conv2d_16 (Conv2D)	(None, 14, 14, 256)	
batch_normalization_7 (BatchNormalization)	(None, 14, 14, 256)	
max_pooling2d_10 (MaxPooling2D)	(None, 6, 6, 256)	
conv2d_17 (Conv2D)	(None, 6, 6, 384)	
conv2d_18 (Conv2D)	(None, 6, 6, 384)	
conv2d_19 (Conv2D)	(None, 6, 6, 256)	
max_pooling2d_11 (MaxPooling2D)	(None, 2, 2, 256)	
flatten_3 (Flatten)	(None, 1024)	
dense_9 (Dense)	(None, 4096)	
dropout_6 (Dropout)	(None, 4096)	
dense_10 (Dense)	(None, 4096)	
dropout_7 (Dropout)	(None, 4096)	
dense_11 (Dense)	(None, 7)	

```
Total params: 24,733,767 (94.35 MB)
Trainable params: 24,733,063 (94.35 MB)
Non-trainable params: 704 (2.75 KB)
```

```
model.fit(train_generator, validation_data=val_generator, epochs=4, batch_size=
```

```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 98s 2s/step - accuracy: 0.1538 - loss: 2.0184 -
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 88s 2s/step - accuracy: 0.1232 - loss: 1.9972 -
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 79s 2s/step - accuracy: 0.1674 - loss: 1.9352 -
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.1808 - loss: 1.9148 -
<keras.src.callbacks.history.History at 0x7e70cc52c1d0>
```

```
import tensorflow as tf
from tensorflow.keras import layers, models

def modified_alexnet(input_shape=(128, 128, 1), num_classes=7):
    model = models.Sequential()

    # First Convolutional Layer (Modified for 1-Channel Input)
    model.add(layers.Conv2D(96, kernel_size=(11, 11), strides=(4, 4), activatic
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Second Convolutional Layer
    model.add(layers.Conv2D(256, kernel_size=(5, 5), activation='relu', padding
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Third, Fourth, and Fifth Convolutional Layers
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(256, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Flattening
    model.add(layers.Flatten())

    # Fully Connected Layers
    model.add(layers.Dense(4505, activation='relu'))
    model.add(layers.Dropout(0.5))
    model.add(layers.Dense(4505, activation='relu'))
    model.add(layers.Dropout(0.5))

    # Output Layer (Change to 8 classes for shape classification)
    model.add(layers.Dense(num_classes, activation='softmax'))

    return model
```

```
# Create model
model = modified_alexnet()
model.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.001, momentum=0
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
Model: "sequential_5"
```

Layer (type)	Output Shape
conv2d_25 (Conv2D)	(None, 30, 30, 96)
batch_normalization_10 (BatchNormalization)	(None, 30, 30, 96)
max_pooling2d_15 (MaxPooling2D)	(None, 14, 14, 96)
conv2d_26 (Conv2D)	(None, 14, 14, 256)
batch_normalization_11 (BatchNormalization)	(None, 14, 14, 256)
max_pooling2d_16 (MaxPooling2D)	(None, 6, 6, 256)
conv2d_27 (Conv2D)	(None, 6, 6, 384)
conv2d_28 (Conv2D)	(None, 6, 6, 384)
conv2d_29 (Conv2D)	(None, 6, 6, 256)
max_pooling2d_17 (MaxPooling2D)	(None, 2, 2, 256)
flatten_5 (Flatten)	(None, 1024)
dense_15 (Dense)	(None, 4505)
dropout_10 (Dropout)	(None, 4505)
dense_16 (Dense)	(None, 4505)
dropout_11 (Dropout)	(None, 4505)
dense_17 (Dense)	(None, 7)

```
Total params: 28,674,073 (109.38 MB)
Trainable params: 28,673,369 (109.38 MB)
Non-trainable params: 704 (2.75 KB)
```

```
model.fit(train_generator, validation_data=val_generator, epochs=4, batch_size=
```

```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 80s 2s/step - accuracy: 0.1345 - loss: 2.1322 -
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.1869 - loss: 1.9715 -
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 86s 2s/step - accuracy: 0.1424 - loss: 1.9467 -
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 134s 2s/step - accuracy: 0.1872 - loss: 1.9023 -
<keras.src.callbacks.history.History at 0x7e70cc50c1d0>
```

```
import tensorflow as tf
from tensorflow.keras import layers, models

def modified_alexnet(input_shape=(128, 128, 1), num_classes=7):
    model = models.Sequential()

    # First Convolutional Layer (Modified for 1-Channel Input)
    model.add(layers.Conv2D(96, kernel_size=(11, 11), strides=(4, 4), activatic
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Second Convolutional Layer
    model.add(layers.Conv2D(256, kernel_size=(5, 5), activation='relu', padding
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Third, Fourth, and Fifth Convolutional Layers
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(256, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Flattening
    model.add(layers.Flatten())

    # Fully Connected Layers
    model.add(layers.Dense(4505, activation='relu'))
    model.add(layers.Dropout(0.5))
    model.add(layers.Dense(4505, activation='relu'))
    model.add(layers.Dropout(0.5))

    # Output Layer (Change to 8 classes for shape classification)
    model.add(layers.Dense(num_classes, activation='softmax'))

    return model
```

```
# Create model
model = modified_alexnet()
model.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
Model: "sequential_6"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━
┃ Layer (type)                          ┃ Output Shape                 ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━
│ conv2d_30 (Conv2D)                    │ (None, 30, 30, 96)           │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ batch_normalization_12                │ (None, 30, 30, 96)           │
│ (BatchNormalization)                  │                              │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ max_pooling2d_18 (MaxPooling2D)       │ (None, 14, 14, 96)           │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ conv2d_31 (Conv2D)                    │ (None, 14, 14, 256)          │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ batch_normalization_13                │ (None, 14, 14, 256)          │
│ (BatchNormalization)                  │                              │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ max_pooling2d_19 (MaxPooling2D)       │ (None, 6, 6, 256)            │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ conv2d_32 (Conv2D)                    │ (None, 6, 6, 384)            │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ conv2d_33 (Conv2D)                    │ (None, 6, 6, 384)            │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ conv2d_34 (Conv2D)                    │ (None, 6, 6, 256)            │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ max_pooling2d_20 (MaxPooling2D)       │ (None, 2, 2, 256)            │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ flatten_6 (Flatten)                   │ (None, 1024)                 │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ dense_18 (Dense)                      │ (None, 4505)                 │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ dropout_12 (Dropout)                  │ (None, 4505)                 │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ dense_19 (Dense)                      │ (None, 4505)                 │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ dropout_13 (Dropout)                  │ (None, 4505)                 │
├───────────────────────────────────────┼──────────────────────────────┼─────
│ dense_20 (Dense)                      │ (None, 7)                    │
└───────────────────────────────────────┴──────────────────────────────┴─────
 Total params: 28,674,073 (109.38 MB)
 Trainable params: 28,673,369 (109.38 MB)
 Non-trainable params: 704 (2.75 KB)
```

```
model.fit(train_generator, validation_data=val_generator, epochs=4, batch_size=
```

```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 80s 2s/step - accuracy: 0.1349 - loss: 2.2799 -
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.1700 - loss: 1.9390 -
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.1516 - loss: 1.9244 -
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 77s 2s/step - accuracy: 0.1800 - loss: 1.8931 -
<keras.src.callbacks.history.History at 0x7e70c193bf50>
```

```
import tensorflow as tf
from tensorflow.keras import layers, models

def modified_alexnet(input_shape=(128, 128, 1), num_classes=7):
    model = models.Sequential()

    # First Convolutional Layer (Modified for 1-Channel Input)
    model.add(layers.Conv2D(96, kernel_size=(11, 11), strides=(4, 4), activatic
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Second Convolutional Layer
    model.add(layers.Conv2D(256, kernel_size=(5, 5), activation='sigmoid', padc
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Third, Fourth, and Fifth Convolutional Layers
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='leaky_relu', p
    model.add(layers.Conv2D(256, kernel_size=(3, 3), activation='tanh', padding
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Flattening
    model.add(layers.Flatten())

    # Fully Connected Layers
    model.add(layers.Dense(4505, activation='relu'))
    model.add(layers.Dropout(0.5))
    model.add(layers.Dense(4505, activation='relu'))
    model.add(layers.Dropout(0.5))

    # Output Layer (Change to 8 classes for shape classification)
    model.add(layers.Dense(num_classes, activation='softmax'))

    return model
```

```
# Create model
model = modified_alexnet()
model.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.001, momentum=0
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
Model: "sequential_7"
```

Layer (type)	Output Shape
conv2d_35 (Conv2D)	(None, 30, 30, 96)
batch_normalization_14 (BatchNormalization)	(None, 30, 30, 96)
max_pooling2d_21 (MaxPooling2D)	(None, 14, 14, 96)
conv2d_36 (Conv2D)	(None, 14, 14, 256)
batch_normalization_15 (BatchNormalization)	(None, 14, 14, 256)
max_pooling2d_22 (MaxPooling2D)	(None, 6, 6, 256)
conv2d_37 (Conv2D)	(None, 6, 6, 384)
conv2d_38 (Conv2D)	(None, 6, 6, 384)
conv2d_39 (Conv2D)	(None, 6, 6, 256)
max_pooling2d_23 (MaxPooling2D)	(None, 2, 2, 256)
flatten_7 (Flatten)	(None, 1024)
dense_21 (Dense)	(None, 4505)
dropout_14 (Dropout)	(None, 4505)
dense_22 (Dense)	(None, 4505)
dropout_15 (Dropout)	(None, 4505)
dense_23 (Dense)	(None, 7)

```
Total params: 28,674,073 (109.38 MB)
Trainable params: 28,673,369 (109.38 MB)
Non-trainable params: 704 (2.75 KB)
```

```
model.fit(train_generator, validation_data=val_generator, epochs=4, batch_size=
```

```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 81s 2s/step - accuracy: 0.1542 - loss: 1.9936 -
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 82s 2s/step - accuracy: 0.1686 - loss: 1.9536 -
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 139s 2s/step - accuracy: 0.1670 - loss: 1.9465 -
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 77s 2s/step - accuracy: 0.1778 - loss: 1.9161 -
<keras.src.callbacks.history.History at 0x7e70b9217550>
```

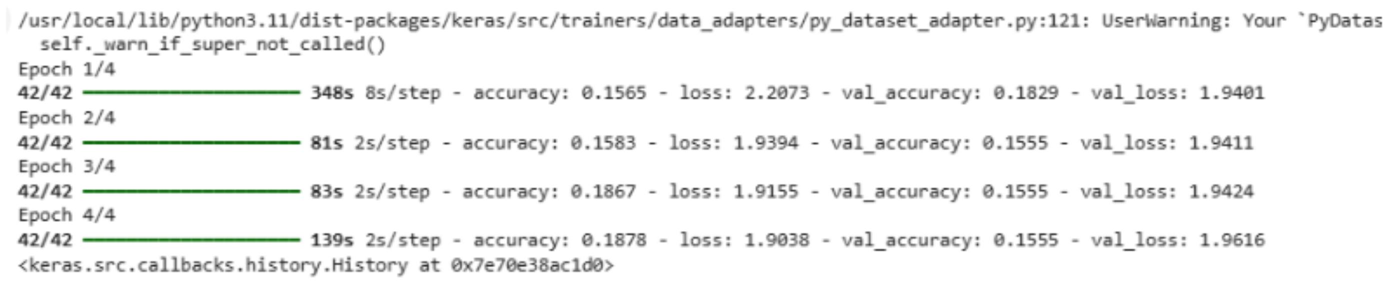
```
import tensorflow as tf
from tensorflow.keras import layers, models

def modified_alexnet(input_shape=(128, 128, 1), num_classes=7):
    model = models.Sequential()

    # First Convolutional Layer (Modified for 1-Channel Input)
    model.add(layers.Conv2D(96, kernel_size=(11, 11), strides=(4, 4), activatic
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Second Convolutional Layer
    model.add(layers.Conv2D(256, kernel_size=(5, 5), activation='relu', padding
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Third, Fourth, and Fifth Convolutional Layers
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.Conv2D(256, kernel_size=(3, 3), activation='relu', padding
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    # Flattening
    model.add(layers.Flatten())

    # Fully Connected Layers
    model.add(layers.Dense(5180, activation='relu'))
    model.add(layers.Dropout(0.5))
    model.add(layers.Dense(5180, activation='relu'))
    model.add(layers.Dropout(0.5))

    # Output Layer (Change to 8 classes for shape classification)
    model.add(layers.Dense(6, activation='softmax'))

    return model
```

```
# Create model
model = modified_alexnet()
model.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.001, momentum=(
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
Model: "sequential_8"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━
┃ Layer (type)                         ┃ Output Shape                ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━
│ conv2d_40 (Conv2D)                   │ (None, 30, 30, 96)          │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ batch_normalization_16               │ (None, 30, 30, 96)          │
│ (BatchNormalization)                 │                             │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ max_pooling2d_24 (MaxPooling2D)      │ (None, 14, 14, 96)          │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ conv2d_41 (Conv2D)                   │ (None, 14, 14, 256)         │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ batch_normalization_17               │ (None, 14, 14, 256)         │
│ (BatchNormalization)                 │                             │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ max_pooling2d_25 (MaxPooling2D)      │ (None, 6, 6, 256)           │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ conv2d_42 (Conv2D)                   │ (None, 6, 6, 384)           │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ conv2d_43 (Conv2D)                   │ (None, 6, 6, 384)           │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ conv2d_44 (Conv2D)                   │ (None, 6, 6, 256)           │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ max_pooling2d_26 (MaxPooling2D)      │ (None, 2, 2, 256)           │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ flatten_8 (Flatten)                  │ (None, 1024)                │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ dense_24 (Dense)                     │ (None, 5180)                │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ dropout_16 (Dropout)                 │ (None, 5180)                │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ dense_25 (Dense)                     │ (None, 5180)                │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ dropout_17 (Dropout)                 │ (None, 5180)                │
├──────────────────────────────────────┼─────────────────────────────┼─────
│ dense_26 (Dense)                     │ (None, 6)                   │
└──────────────────────────────────────┴─────────────────────────────┴─────
 Total params: 35,903,542 (136.96 MB)
 Trainable params: 35,902,838 (136.96 MB)
 Non-trainable params: 704 (2.75 KB)
```

Results:

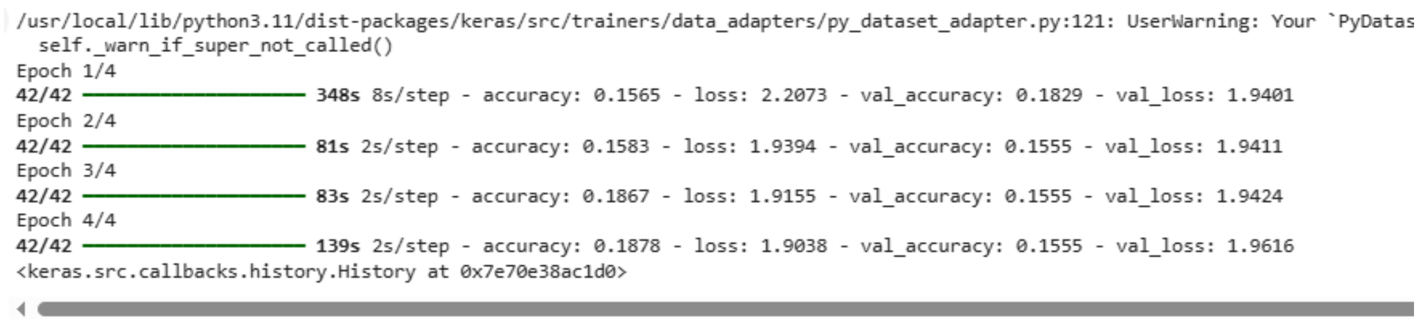
Part 2:



```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDatas
  self._warn_if_super_not_called()
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 348s 8s/step - accuracy: 0.1565 - loss: 2.2073 - val_accuracy: 0.1829 - val_loss: 1.9401
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 81s 2s/step - accuracy: 0.1583 - loss: 1.9394 - val_accuracy: 0.1555 - val_loss: 1.9411
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 83s 2s/step - accuracy: 0.1867 - loss: 1.9155 - val_accuracy: 0.1555 - val_loss: 1.9424
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 139s 2s/step - accuracy: 0.1878 - loss: 1.9038 - val_accuracy: 0.1555 - val_loss: 1.9616
<keras.src.callbacks.history.History at 0x7e70e38ac1d0>
```

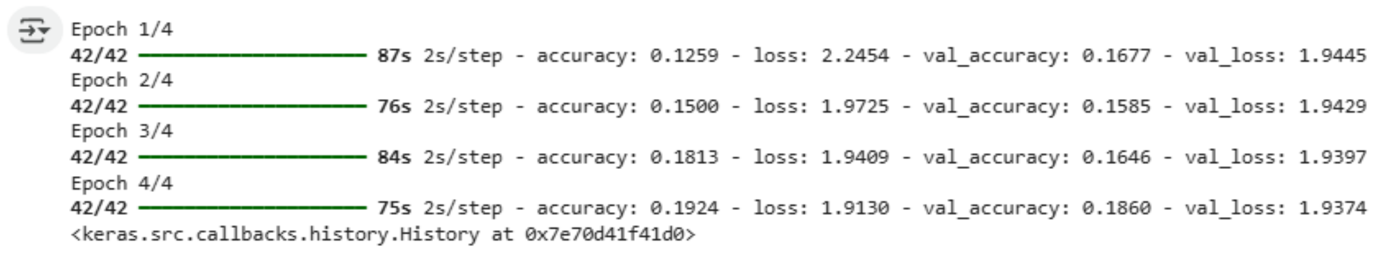
Part 3:

Accuracy with Learning rate 0.01:



```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDatas
  self._warn_if_super_not_called()
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 348s 8s/step - accuracy: 0.1565 - loss: 2.2073 - val_accuracy: 0.1829 - val_loss: 1.9401
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 81s 2s/step - accuracy: 0.1583 - loss: 1.9394 - val_accuracy: 0.1555 - val_loss: 1.9411
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 83s 2s/step - accuracy: 0.1867 - loss: 1.9155 - val_accuracy: 0.1555 - val_loss: 1.9424
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 139s 2s/step - accuracy: 0.1878 - loss: 1.9038 - val_accuracy: 0.1555 - val_loss: 1.9616
<keras.src.callbacks.history.History at 0x7e70e38ac1d0>
```

Accuracy with Learning rate 0.001:



```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 87s 2s/step - accuracy: 0.1259 - loss: 2.2454 - val_accuracy: 0.1677 - val_loss: 1.9445
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.1500 - loss: 1.9725 - val_accuracy: 0.1585 - val_loss: 1.9429
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 84s 2s/step - accuracy: 0.1813 - loss: 1.9409 - val_accuracy: 0.1646 - val_loss: 1.9397
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 75s 2s/step - accuracy: 0.1924 - loss: 1.9130 - val_accuracy: 0.1860 - val_loss: 1.9374
<keras.src.callbacks.history.History at 0x7e70d41f41d0>
```

1st Sigmoid 3rd and 5th tanh for activation functions:

```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 80s 2s/step - accuracy: 0.1290 - loss: 2.0077 - val_accuracy: 0.1555 - val_loss: 1.9424
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 89s 2s/step - accuracy: 0.1584 - loss: 1.9464 - val_accuracy: 0.1555 - val_loss: 1.9440
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 85s 2s/step - accuracy: 0.1549 - loss: 1.9441 - val_accuracy: 0.1555 - val_loss: 1.9465
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 142s 2s/step - accuracy: 0.1925 - loss: 1.9115 - val_accuracy: 0.1098 - val_loss: 1.9492
<keras.src.callbacks.history.History at 0x7e70d42bc1d0>
```

2nd sigmoid 4th leaky_relu 5th tanh for activation functions:

```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 98s 2s/step - accuracy: 0.1538 - loss: 2.0184 - val_accuracy: 0.1098 - val_loss: 1.9530
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 88s 2s/step - accuracy: 0.1232 - loss: 1.9972 - val_accuracy: 0.1341 - val_loss: 1.9461
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 79s 2s/step - accuracy: 0.1674 - loss: 1.9352 - val_accuracy: 0.1555 - val_loss: 1.9437
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.1808 - loss: 1.9148 - val_accuracy: 0.1555 - val_loss: 1.9447
<keras.src.callbacks.history.History at 0x7e70cc52c1d0>
```

Part 4:

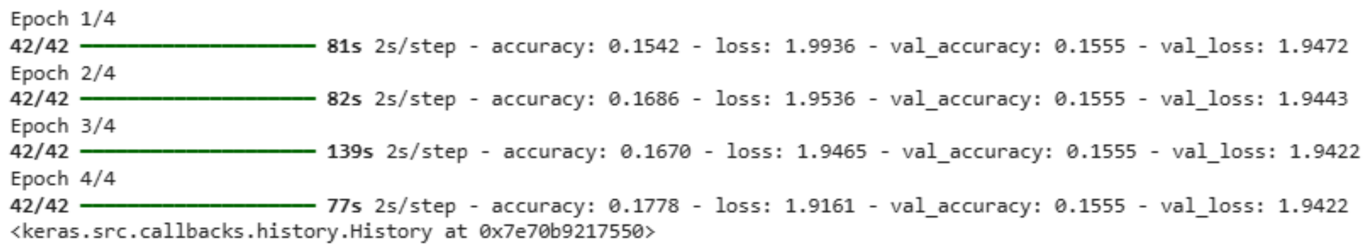
%10 increased fully connected layer nodes (4096-> 4505)

```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 80s 2s/step - accuracy: 0.1345 - loss: 2.1322 - val_accuracy: 0.1585 - val_loss: 1.9451
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.1869 - loss: 1.9715 - val_accuracy: 0.1555 - val_loss: 1.9430
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 86s 2s/step - accuracy: 0.1424 - loss: 1.9467 - val_accuracy: 0.1555 - val_loss: 1.9408
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 134s 2s/step - accuracy: 0.1872 - loss: 1.9023 - val_accuracy: 0.1555 - val_loss: 1.9399
<keras.src.callbacks.history.History at 0x7e70cc50c1d0>
```

Learning rate 0.01:

```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 80s 2s/step - accuracy: 0.1349 - loss: 2.2799 - val_accuracy: 0.1555 - val_loss: 1.9417
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.1700 - loss: 1.9390 - val_accuracy: 0.1524 - val_loss: 1.9495
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.1516 - loss: 1.9244 - val_accuracy: 0.1555 - val_loss: 1.9615
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 77s 2s/step - accuracy: 0.1800 - loss: 1.8931 - val_accuracy: 0.1494 - val_loss: 2.0238
<keras.src.callbacks.history.History at 0x7e70c193bf50>
```

2nd sigmoid 4th leaky_relu 5th tanh for activation functions:



```
Epoch 1/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 81s 2s/step - accuracy: 0.1542 - loss: 1.9936 - val_accuracy: 0.1555 - val_loss: 1.9472
Epoch 2/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 82s 2s/step - accuracy: 0.1686 - loss: 1.9536 - val_accuracy: 0.1555 - val_loss: 1.9443
Epoch 3/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 139s 2s/step - accuracy: 0.1670 - loss: 1.9465 - val_accuracy: 0.1555 - val_loss: 1.9422
Epoch 4/4
42/42 ━━━━━━━━━━━━━━━━━━━━ 77s 2s/step - accuracy: 0.1778 - loss: 1.9161 - val_accuracy: 0.1555 - val_loss: 1.9422
<keras.src.callbacks.history.History at 0x7e70b9217550>
```

Comments and Discussions:

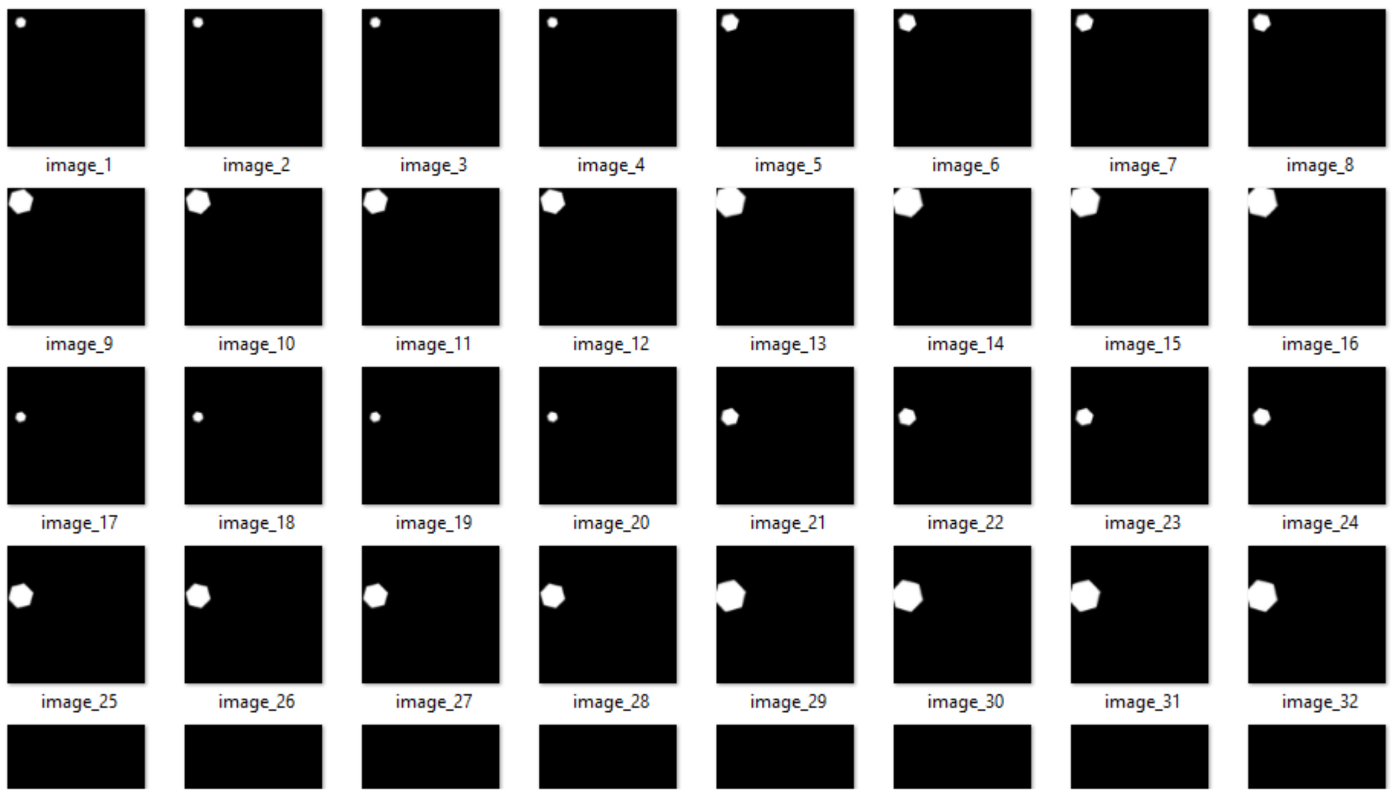
128x128 image generation

I used 2D Shape library to generate image . After generated image was 255x255 and it did not contain noise. So i resize it to 128x128 and added noise to images. I can not find how to generate triangle , I examined the files but I could not find an input generation part related to the triangle, I may have missed it.%80 of data used for training and %20 for testing. Cross validation also helps to optimal model but models takes time to generate.(can't deliver on time properly if i try it)

- My input is 7 because i can not generate triangle so my output is 7 too . i need to indicate it again.
- Example generate shape:

!python main.py --shapes star5 --to_transform trx try scale rota --num_transformation 3 --canvas_bccol 0 0 0 --stim_colour 255 255 255

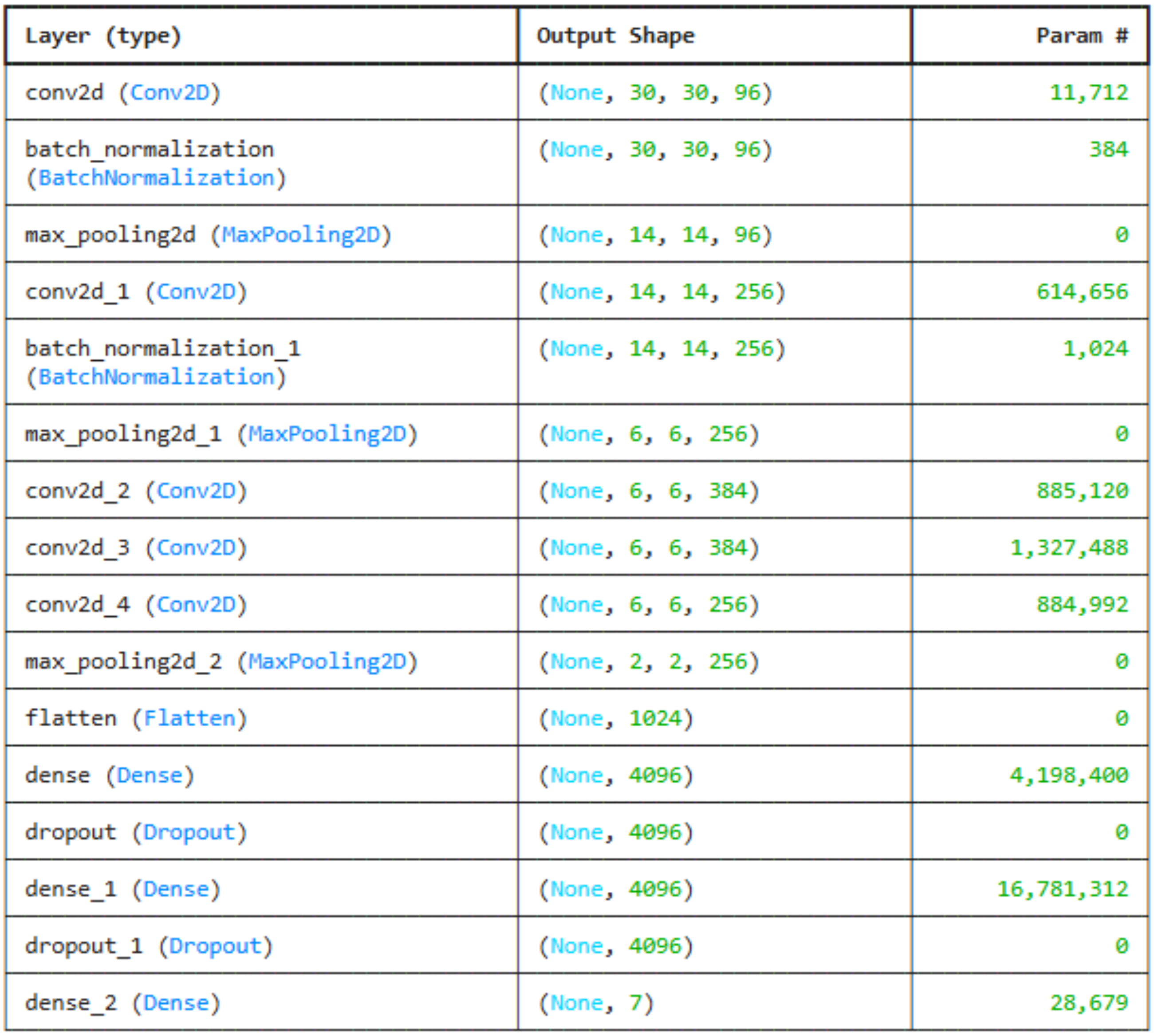
This code generates cartesian product of translation on x, translation on y ,scale and rotation so it generate 3x3x3x3 = 81 images.



```python
import scipy.io
import numpy as np
from PIL import Image

def mat_to_png(mat_file, output_folder, target_size=(128, 128)):
    # Load the .mat file
    mat_data = scipy.io.loadmat(mat_file)

    # Assuming the images are stored in the key 'images' (adjust if necessary)
    images_data = mat_data['images']

    # Extract images from the 5D array
    images = images_data[0]  # Since the first dimension is 1

    # Loop through all 20 images
    for idx, img_data in enumerate(images):
        # Convert the numpy array to a PIL image (assuming the array is RGB)
        img = Image.fromarray(img_data.astype(np.uint8), 'RGB')

        # Resize the image to 128x128
        img_resized = img.resize(target_size, Image.LANCZOS)

        # Save each image as a .png file
        img_resized.save(f"{output_folder}/image_{idx + 1}.png")

# Example usage
mat_to_png('/content/drive/MyDrive/2D-Shape-Generator-master/output/shapes_star5_dims_trx_try_scale_rota.mat', 'star5')
```

AlexNet Model - Part2

After dataset provided , alexnet model generated with below settings.



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 96)	11,712
batch_normalization (BatchNormalization)	(None, 30, 30, 96)	384
max_pooling2d (MaxPooling2D)	(None, 14, 14, 96)	0
conv2d_1 (Conv2D)	(None, 14, 14, 256)	614,656
batch_normalization_1 (BatchNormalization)	(None, 14, 14, 256)	1,024
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 256)	0
conv2d_2 (Conv2D)	(None, 6, 6, 384)	885,120
conv2d_3 (Conv2D)	(None, 6, 6, 384)	1,327,488
conv2d_4 (Conv2D)	(None, 6, 6, 256)	884,992
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 256)	0
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 4096)	4,198,400
dropout (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 4096)	16,781,312
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 7)	28,679

Model Results : accuracy: 0.1878 - loss: 1.9038 - val_accuracy: 0.1555 - val_loss: 1.9616

- It has too low accuracy %15-18.
- I used really low epoch values because it takes times to train.(wouldn't have made it on time.)

Part 3:

- Learning rate 0.001 created more accurate model %18-19
- Learning rate 0.001 is better between i used so i choose it.
- First relu changed as sigmoid also third and fifth convolutional layers changed as tanh. Accuracy is still validation accuracy is terrible it drops to %10's (probably underfit but seems like overfit because train_validation still %19 even this bad results).
- 2nd sigmoid 4th leaky_relu 5th tanh for activation functions and this model accuracy still between %15-18.
- Sigmoid , tanh , relu , leaky relu are really good activation functions for CNN architecture. Different combinations tried and i will continue with based alexnet architecture.

Part 4:

- Fully connected layer nodes increased %10 and 4096 to 4505.
- Although the accuracy decreased, the validation loss also decreased, but I think that increasing the model complexity will increase the validation loss as in the bias-variance trade-off graph.
- Learning rate 0.01 which is not selected gave similar result accuracy is decreased to %18.0 and also validation lose increased.
- After tried with different activation functions accuracy rate is not increase also validation error increase.
- Then fully connected layer increased %15 and output layer 1 decreased.
- The last part of removing layer can not handled correct way . Probably accuracy will increase because class count decrease even complexity increase.