Feed-forward Neural Network

A Feed-forward Neural Network (FNN), is a fundamental type of artificial neural network or ANN. It is widely used in various machine learning tasks which include classification, regression, and pattern recognition. The name Feed-Forward comes from the fact that the flow of information is one-directional - from the input layer, through one or more hidden layers, to the output layer.

Structure:

- **Input Layer:** consists of the neurons that represent the input features of the data. Each neuron corresponds to a feature, and the number of neurons is equal to the number of input features.
- **Hidden Layers:** are one or multiple layers between the input and output layers. Each hidden layer contains neurons that transform the input data through weighted sums and activation functions. The number of hidden layers and neurons in each layer depends on the complexity of the problem and computational resources.
- **Output Layer:** produces the final output of the network. The number of neurons in the output layer depends on the nature of the task.

Neurons and Activation Functions:

Each neuron computes a weighted sum of inputs, adds bias, and applies an activation function like sigmoid, tanh, ReLU, or softmax to introduce non-linearity, enabling complex pattern learning.

Training:

During training, the network adjusts weights and biases using optimization algorithms like gradient descent to minimize a loss function. Backpropagation efficiently computes gradients of the loss function with respect to weights for updating them.

Regularization and Optimization:

Techniques like dropout, weight decay, and batch normalization prevent overfitting and improve generalization. Optimization algorithms like Adam, RMSprop, and SGD with momentum efficiently update weights during training.

Evaluation:

Once trained, the performance of the FNN is evaluated on unseen data using metrics appropriate for the task, such as accuracy, precision, recall, F1-score, or Mean Squared Error (MSE) for regression tasks.

Dataset:

The dataset was obtained using this python code:

```
import pandas as pd
import numpy as np

# Generate synthetic data
np.random.seed(42)
num_samples = 10000

# Generate random features
data = {
    'source_ip': [f'192.168.{np.random.randint(1,
255)}.{np.random.randint(1, 255)}' for _ in range(num_samples)],
    'dest_ip': [f'10.0.{np.random.randint(1,
255)}.{np.random.randint(1, 255)}' for _ in range(num_samples)],
    'protocol': np.random.choice(['TCP', 'UDP', 'ICMP'],
size=num_samples),
    'packet_size': np.random.randint(64, 1500, size=num_samples),
    'num_packets': np.random.randint(1, 100, size=num_samples),
    'duration': np.random.uniform(0.1, 10, size=num_samples),
    'label': np.random.choice(['normal', 'intrusion'],
size=num_samples)
}

# Create DataFrame
df = pd.DataFrame(data)

# Save DataFrame to CSV
df.to_csv('network_traffic_data.csv', index=False)

print("Dataset created and saved successfully.")
```

Here are some rows from the dataset:

source_ip	dest_ip	protocol	packet_size	num_packets	duration	label
192.168.103.180	10.0.132.86	UDP	1153	58	9.703647888 24104	intrusion
192.168.93.15	10.0.243.183	ICMP	258	1	9.011383354 974434	intrusion
192.168.107.	10.0.231.228	TCP	1275	22	5.719440251	intrusion

72					882002	
192.168.189.21	10.0.162.27	ICMP	110	95	7.8346101694851455	normal
192.168.103.122	10.0.7.131	ICMP	1449	23	9.105900735420462	intrusion
192.168.211.215	10.0.175.225	UDP	1072	8	2.511642610199937	normal
192.168.75.203	10.0.2.234	TCP	600	29	5.836870383595166	normal
192.168.88.117	10.0.80.153	ICMP	910	44	4.868991560600228	normal
192.168.100.104	10.0.60.207	UDP	772	62	7.73875441294512	normal
192.168.152.131	10.0.24.15	UDP	1333	19	3.852916961204904	normal
192.168.150.53	10.0.95.142	UDP	871	59	7.118898433111098	intrusion
192.168.2.88	10.0.182.206	UDP	675	80	4.089631581216204	intrusion
192.168.236.158	10.0.134.206	UDP	1265	19	1.1996411599805918	normal
192.168.38.130	10.0.81.18	ICMP	178	22	1.1605021729223406	intrusion
192.168.192.188	10.0.20.115	ICMP	743	83	1.2786576130765552	normal
192.168.21.161	10.0.190.209	TCP	694	68	0.42194677172840156	normal
192.168.204.58	10.0.208.40	ICMP	1252	5	5.376434398808043	intrusion
192.168.22.253	10.0.39.180	TCP	282	5	6.751836551638975	normal
192.168.236.89	10.0.32.124	TCP	260	25	1.3546111780743268	intrusion
192.168.49.219	10.0.108.201	TCP	464	24	6.1573942021939585	normal
192.168.59.170	10.0.27.110	ICMP	812	1	6.017626774753328	intrusion
192.168.220.188	10.0.180.97	UDP	744	28	1.5562311088289378	normal