

```
class NeuralNetwork(input_size, output_size, hidden_layers_sizes,
activation
```

Parameters

- **input_size**: Number of features in an input
- **output_size**: Number of output classes
- **hidden_layer_sizes**: list, The ith element represents the number of units (perceptrons) in the ith hidden layer.
- **activation_funtion**: Type of nonlinearity to be used for every perceptron except the output_layer.
 - logistic: the logistic sigmoid function, returns $f(x) = 1 / (1 + \exp(-x))$.
 - tanh: the hyperbolic tan function, returns $f(x) = \tanh(x)$.
 - relu: the rectified linear unit function, returns $f(x) = \max(0, x)$

Note: Always use sigmoid as the non linearity for the output layer.

For e.g `NeuralNetwork(500, 2, [100, 10])` represents a neural network work that takes in an input of size 500 and classifies it into one of the two output_categories. It consist of two hidden layers with 100 and 10 pereptrons respectively.

Functions

1. `train(X, y, batch_size, η_0 , max_iterations)`

Parameters

- **X**: the input data, shape (n_samples, n_features)
- **y**: The target values (class labels in classification)
- **batch_size**: batch size for sgd
- **η_0** : initial learning rate
- **max_interation**: maximum number of iterations

2. `predict(X)`: predict using the model

Parameters - **X**: the input data, shape (n_samples, n_features)