Week 9

Multilayer Perceptron (MLP)

- Supervised learning algorithm
- MLP learns a **non-linear function approximator** for either classification or regression depending on the given dataset.
- In sklearn, we implement MLP using:
    - MLPClassifier for classification
    - MLPRegressor for regression
- MLPClassifier supports multi-class classification by applying Softmax as the output function.
- It also supports multi-label classification.
- MLPRegressor also supports multi-output regression, in which a sample can have more than one target.

MLPClassifier - Implementation

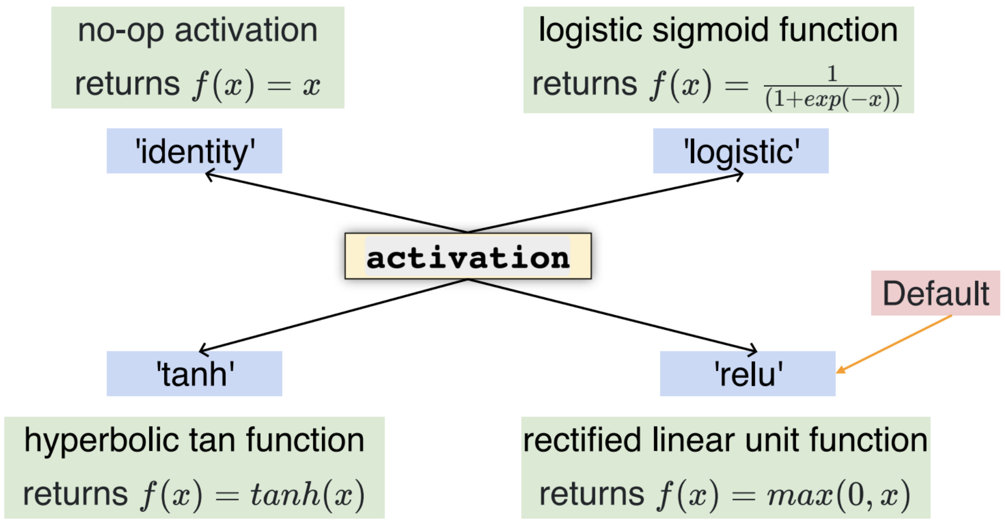
```
from sklearn.neural_network import MLPClassifier

MLP_clf = MLPClassifier()
MLP_clf.fit(X_train, y_train)

MLP_clf.predict(X_test)
#gives a vector of probability estimates per sample
MLP_clf.predict_proba(X_test)
```

Parameters

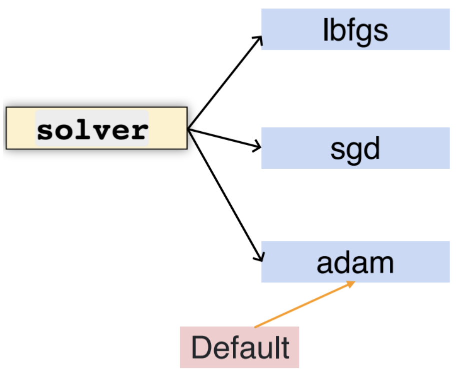
- **hidden_layer_sizes:**
    - This parameter sets the number of layers and the number of neurons in each layer.
    - It is a tuple where each element in the tuple represents the number of neurons at the ith position where i is the index of the tuple.
    - The length of tuple denotes the total number of hidden layers in the network.

```
MLPClassifier(hidden_layer_sizes=(15,10,5))
```

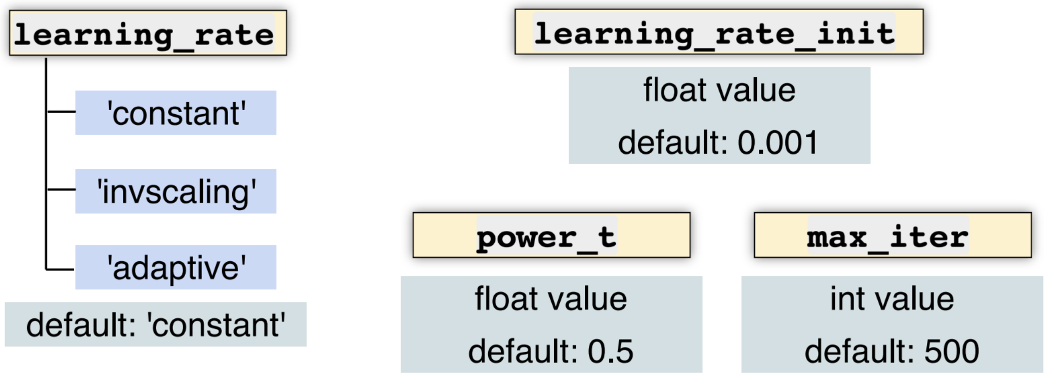
- **alpha (regularisation):**
    - Strength of the L2 regularization term.
    - Default: alpha = 0.0001
- **activation function (for hidden layers):**

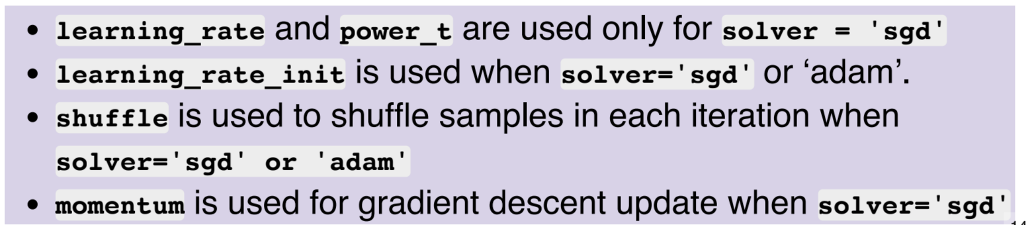


- **solver (weight optimisation):**



- **batch_size:**
  - If the solver is ‘lbfgs’, the classifier will not use minibatch.
  - Size of minibatches can be set to other stochastic optimizers: batch_size (int)
  - default batch_size is 'auto'.
  - `batch_size=min(200, n_samples)`




- `learning_rate` and `power_t` are used only for `solver = 'sgd'`
- `learning_rate_init` is used when `solver='sgd'` or ‘adam’.
- `shuffle` is used to shuffle samples in each iteration when `solver='sgd' or 'adam'`
- `momentum` is used for gradient descent update when `solver='sgd'`

Attributes

- **coefs_ (weight matrix coefficients):**
  - It is a list of shape (n_layers - 1,)
  - The ith element in the list represents the weight matrix corresponding to layer i.

- **intercepts_ (bias vector):**
  - It is a list of shape (n_layers - 1,)
  - The ith element in the list represents the weight matrix corresponding to layer i.

MLPRegressor

- MLPRegressor trains using backpropagation with no activation function in the output layer.
- Therefore, it uses the square error as the loss function, and the output is a set of continuous values.

Clustering

Hierarchical Agglomerative Clustering (HAC)

Approaches

- Hierarchial clustering starts by considering each datum as a cluster and then combines closest clusters to form larger clusters. This is bottoms up approach.
- There is an alternate approach, which is top-down approach, where the entire data is considered as a one single cluster, which is divided to form smaller clusters in each step.

Algorithm

1. Calculate the distance matrix between pairs of clusters.
2. while all the objets are clustered into one.

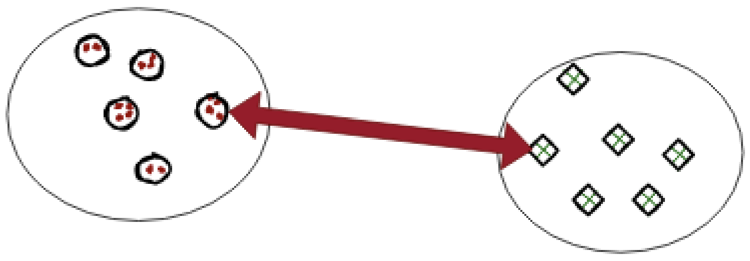
   2a. Detect the two closest groups (clusters) and merge them.

Linkage

Linkage is a strategy for aggregating clusters.

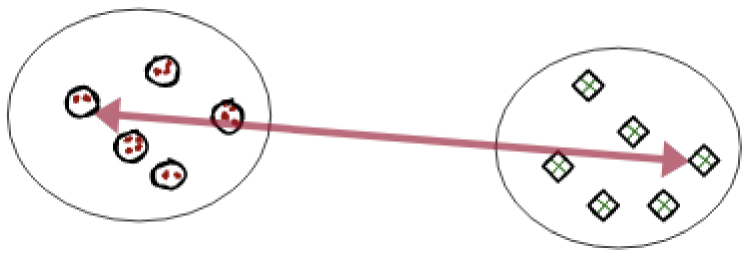
- **Single Linkage**

  merges clusters based on the shortest distance over all possible pairs



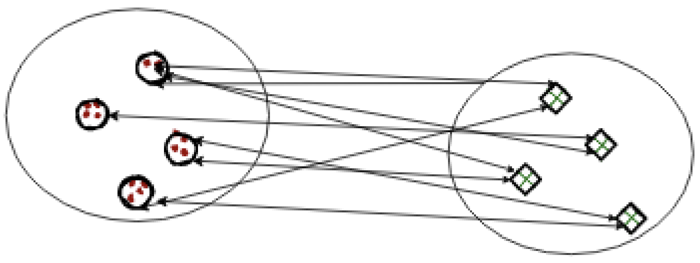
- **Complete Linkage**

  merges clusters to minimize the maximum distance between the clusters



- **Average Linkage**

  uses average distance over all possible pairs between the groups for merging clusters



- **Ward’s Linkage**

  computes the sum of squared distances within the clusters