

# Toolkit

Module: `deeprai.tools.toolkit`

This module provides a collection of utility functions designed for numpy arrays. These functions offer various operations like verification, rounding, normalization, reshaping, and others, enhancing usability and information retrieval from numpy arrays.

---

## 1. `verify_inputs(array)`

### Description:

Verify if the given input is a numpy array.

### Parameters:

- **array**: The input to be checked.

### Returns:

- **bool**: True if the input is a numpy array, otherwise False.

### Example:

```
from deeprai.tools.toolkit import verify_inputs

result = verify_inputs(np.array([1, 2, 3]))
print(result) # True
```

---

## 2. `round_out(array, a=2)`

### Description:

Round the elements of a numpy array and set specific print options.

### Parameters:

- **array** (`np.ndarray`): The input numpy array.
- **a** (`int`, optional): Decimal places to round to. Defaults to 2.

### Returns:

- **np.ndarray**: The rounded numpy array.

### Example:

```
from deeprai.tools.toolkit import round_out

rounded_array = round_out(np.array([1.12345, 2.6789]))
print(rounded_array) # [1.12, 2.68]
```

## 3. `normalize(array)`

### Description:

Normalize the elements of the numpy array to the range [0, 1].

### Parameters:

- **array** (`np.ndarray`): The input array.

## Returns:

- **np.ndarray**: The normalized array.

## Example:

```
from deeprai.tools.toolkit import normalize

norm_array = normalize(np.array([10, 20, 30, 40]))
print(norm_array)
```

---

## 4. `reshape_to_2d(array)`

## Description:

Reshape the numpy array to a 2D format if it's not already in that shape.

## Parameters:

- **array** (`np.ndarray`): The input array.

## Returns:

- **np.ndarray**: The reshaped 2D array.

## Example:

```
from deeprai.tools.toolkit import reshape_to_2d

reshaped_array = reshape_to_2d(np.array([1, 2, 3, 4]))
print(reshaped_array)
```

---

## 5. `is_square_matrix(array)`

### Description:

Check if the given numpy array is a square matrix.

### Parameters:

- **array** (`np.ndarray`): The input array.

### Returns:

- **bool**: True if the array is a square matrix, otherwise False.

### Example:

```
from deeprai.tools.toolkit import is_square_matrix

result = is_square_matrix(np.array([[1, 2], [3, 4]]))
print(result) # True
```

## 6. `sum_along_axis(array, axis=0)`

### Description:

Compute the sum of elements of the numpy array along a specified axis.

### Parameters:

- **array** (`np.ndarray`): The input array.
- **axis** (`int`, optional): Axis along which the sum is computed. Defaults to 0.

# Returns:

- **np.ndarray**: The sum along the specified axis.

# Example:

```
from deeprai.tools.toolkit import sum_along_axis

summed_array = sum_along_axis(np.array([[1, 2], [3, 4]]))
print(summed_array) # [4, 6]
```

## 7. `array_info(array)`

# Description:

Retrieve essential information about the numpy array.

# Parameters:

- **array** (`np.ndarray`): The input array.

# Returns:

- **dict**: A dictionary containing shape, data type, minimum and maximum values.

# Example:

```
from deeprai.tools.toolkit import array_info

info = array_info(np.array([[1, 2], [3, 4]]))
print(info)
```