

Data

An interface for data that should be stored in memory. The interface is comprised of 3 methods: [getData](#), [getPath](#) and [isValid](#).

Network Data

An implementation of Data, contains path, inputSize, layers, iterations, activation and targetSet. Used to store data about the neural network.

```
$networkData = new NetworkData;
$networkData->path;
// returns "network//meta.txt"
```

Storage Data

An implementation of Data, contains path, wordCount, courses, material and trainingPercentage. Used to store data about the samples.

```
$storageData = new StorageData;
$storageData->path;
// returns "samples//meta.txt"
```

Get Data

To get raw data of an object implementing the Data interface, use `getData`. The result is conventionally the data contained in the objects properties merged into a string separated by new-lines for the purpose of writing to memory.

```
$storageData = new StorageData;
$storageData->wordCount = 10;
$storageData->course = ["906","968","1026"];
$storageData->material = ["pdfs"];
$storageData->trainingPercentage = 20;
$storageData->getData();
// returns "10\n906,968,1026\npdfs\n20"
```

Get Path

To get the path to storage location of an object implementing the Data interface, use `getPath`.

```
$storageData = new StorageData;
$storageData->getPath();
// returns "samples//meta.txt"
```

Is Valid

To determine whether an object implementing the Data interface has valid properties, use `isValid`.

```
$storageData = new StorageData;
$storageData->isValid();
```

```
// returns 'false' due to null properties
```