Predictive maintenance Project

implemented on .Net C#

XisomDeepLearning Document

.Net API

Design deep model

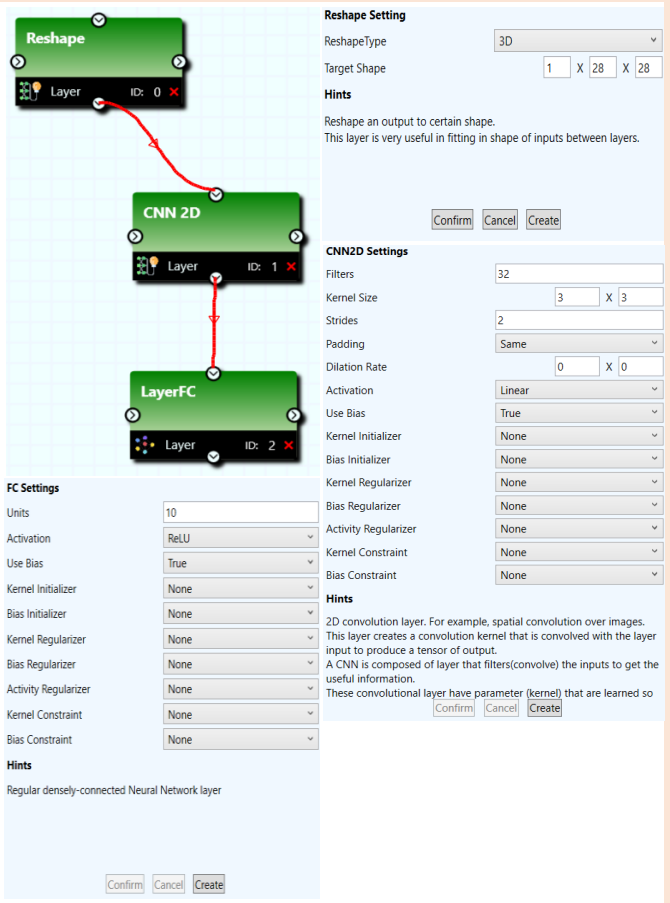
XisomDeepLearning provides high-level neural network APIs to develop and evaluate deep learning models

```
using XisomNet.Layers;
using XisomNet.Layers.Activations;

var model = new Sequential();
model.Add(new Reshape(new long[] { 1, 28, 28 }));
model.Add(new Conv2D(filters: 32,
                     kernelSize: Tuple.Create(3, 3),
                     strides: 2,
                     padding: PaddingType.Same,
                     useBias: true));
model.Add(new Dense(units: 10,
          activation:  ActiveFunctionType.ReLU,
          useBias: true));
```

GUI Tool

Design deep model



.Net API

Preprocessing

Prepare the data for training model. It includes partitioning dataset into training data, testing data and preprocessing tasks such as: Denoising, augmentation, normalization.

Normalization

```
var MinMaxScaller = new MinMaxScaller();
rawDatas = MinMaxScaller.FitTransform(rawDatas);
rawTestDatas = MinMaxScaller.Transform(rawTestDatas);
```

Forming data

```
var XY = PrepDatasetTimeSeries(rawDatas, 50000);
var TestXY = PrepDatasetTimeSeries(rawTestDatas, 50000);
var DataSet = new DataFrameXY(XY.Item1, XY.Item2);
var TestSet = new DataFrameXY(TestXY.Item1, TestXY.Item2);
```

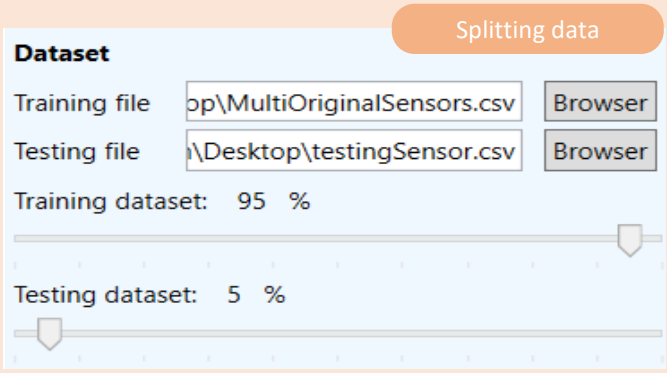
Splitting data

```
var TrainVal = DataSet.SplitData(0.05);
```

GUI Tool

Preprocessing

Splitting data



.Net API

Compile model

Config the model for training

Binary classification

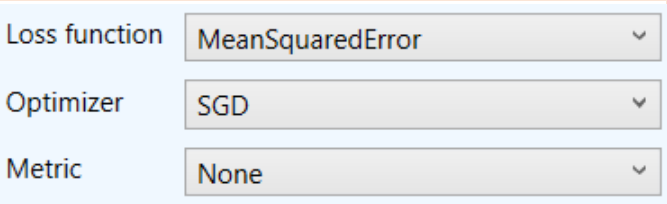
```
model.Compile(OptimizerType.SGD,
              LossFunctionType.BinaryCrossEntropy,
              MetricType.Accuracy);
```

Multi-classification

```
model.Compile(OptimizerType.Adam,
              LossFunctionType.CategorialCrossEntropy,
              MetricType.Accuracy);
```

GUI Tool

Compile model



XisomDeepLearning Document

.Net API

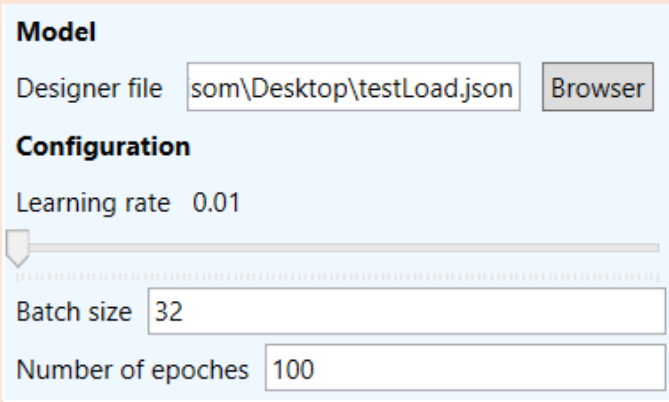
Training model

Use previous configuration and setting parameters to optimize model

```
model.Train(TrainVal.Item1, epochs: 100, batchSize: 16,
            TrainVal.Item2);
```

GUI Tool

Training model



.Net API

Save/Load model

Save/load the trained model into/from hard disk for storage / performing with new data.

Save trained model

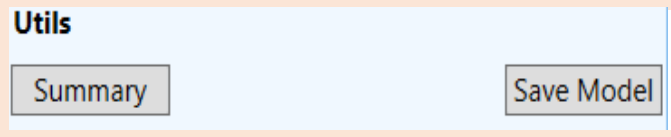
```
model.SaveModel(@".\XisomModel.json");
```

Load trained model

```
model.LoadModel(@".\XisomModel.json");
```

GUI Tool

Inspecting model



.Net API

Early stopping

This algorithm is used to stop training when the improvement is not increased.

Until last epoch

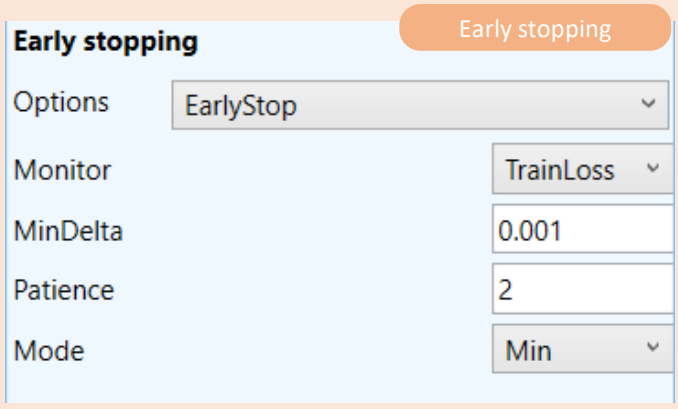
```
model.Train(trainSet, epoches, batchsize);
```

Early stopping

```
EarlyStopping es = new EarlyStopping(MonitorType, MinDelta, Patience,
MonitorMode);
MainModel.Train(trainSet, epoches, batchsize, null, es);
```

.Net API

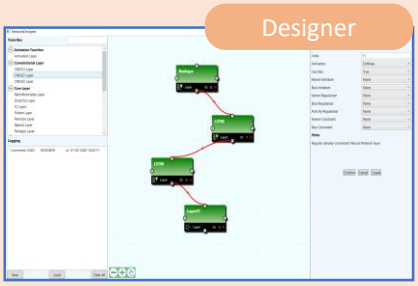
Early stopping



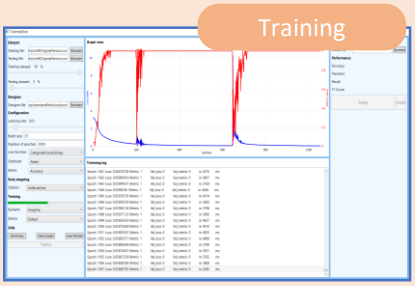
Workflows.

Methodology of developing a machine learning model.

Designer



Training



Demonstration