The Machine Learning Package
Nov. 2023, Max Kleiner

- MLP as Package with out of the box demos
- IDE / RAD / CLI / Shell / Scripts
- CIFAR-10 Image Classifier 
- https://github.com/maxkleiner/neural-api
- This session shows you various ways of using the MLP in your application.

maXbox4exe

Digital unterschrieben
von maXbox4exe
Datum: 2023.08.15
17:45:46 +02'00'

EKON 27

Agenda

- Artificial Neural Networks (ANNs) in a Delphi Form as a Component.
- ML Package for Delphi and Lazarus.
- CIFAR-10 Image Classifier
- Loading and testing a pre-trained model
- *In the archive **MachineLearningPackage.zip** you find the script, model and data you need, which works with Lazarus, Delphi, Jupyter and maXbox.*

NN Research

- Neural networks are composed of a large number of interconnected units divided into input, output, and hidden nodes. A single processing unit merely sums up the weighted activation on its inputs, transforms this sum according to an activation function, and passes the resulting function to its output.
- https://www.kau.edu.sa/Files/320/Researches/52692_22998.pdf
- https://entwickler-konferenz.de/blog/machine-learning-mit-cai/

Cross-platform ANN component

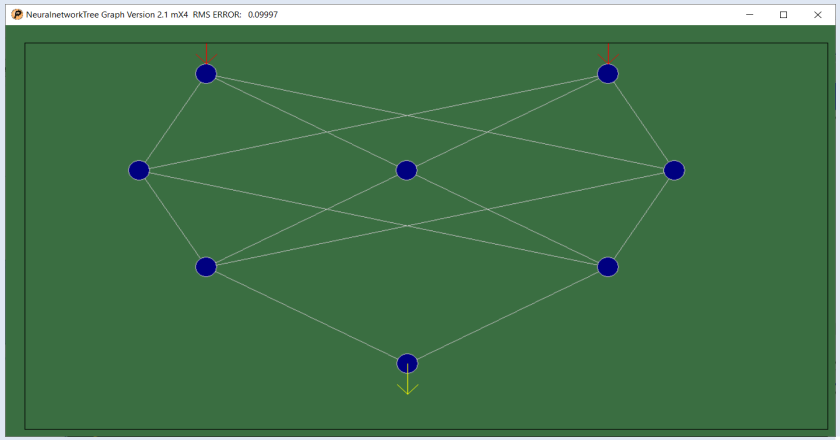
- Defining the Network Structure.
- Initialization (structure).
- Specifying minimum and maximum values for inputs and outputs.
- Training data (with loss function).
- After Training testing.
- Using the Trained Network

https://github.com/maxkleiner/maXbox/blob/master/logisticregression2.ipynb

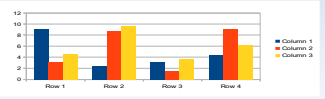
EKON 27

Defining NN Structure

- To define structure of a network run-time, just add following lines:
- nn1.Network.clear;
- nn1.Network.Add('2'); // Number of inputs
- nn1.Network.Add('3'); // Number of hidden neurons
- nn1.Network.Add('2');
- nn1.Network.Add('1'); // Num. of outputs
- Nn1.Initialize(true); // Initialize neural network



EKON 27



Be aware of

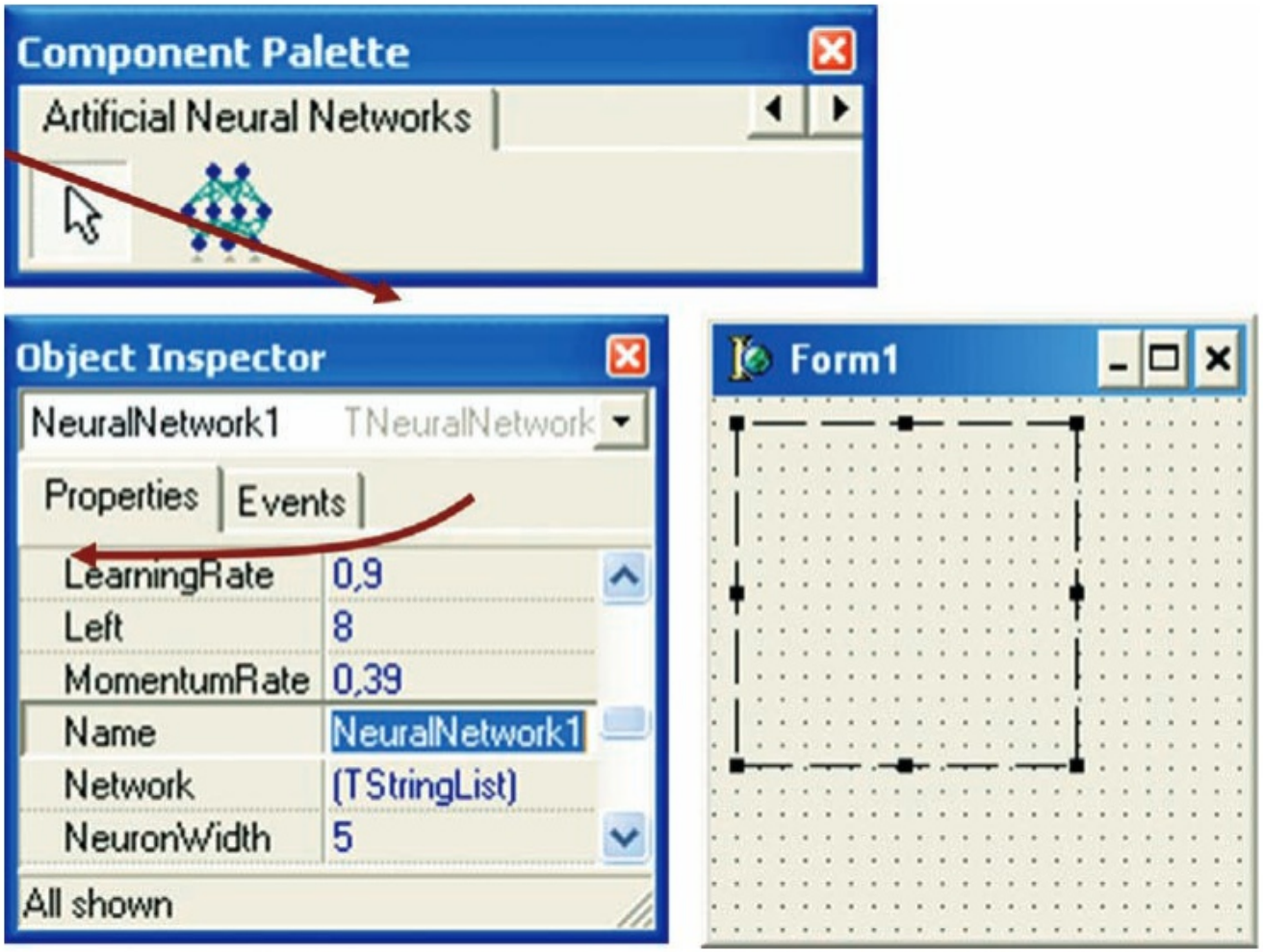
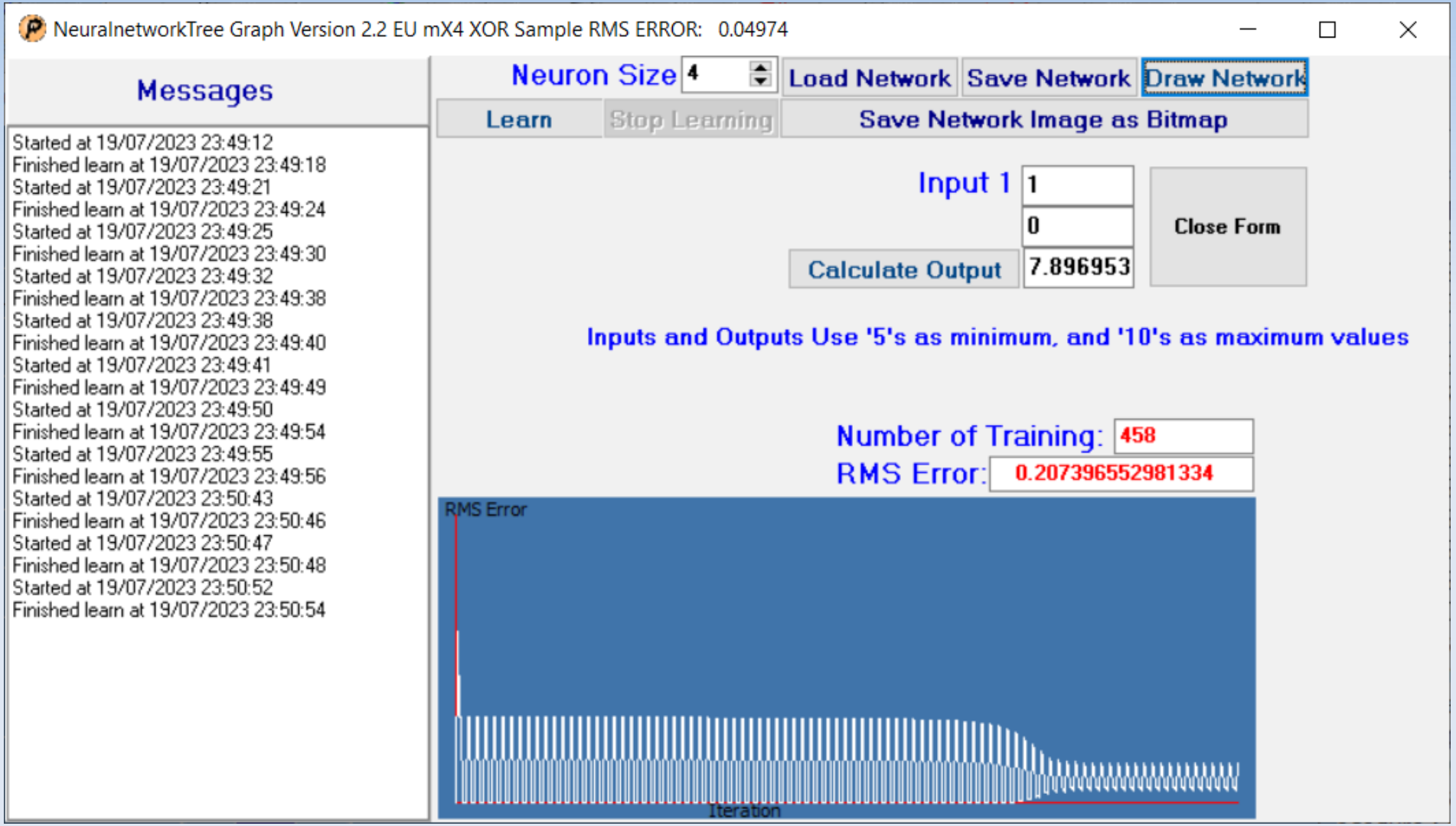


Fig. 4. Adding an instance of ANN component onto design form.

Demo: 1234_NeuralNetwork2_XOR_test12_EKON27.pas

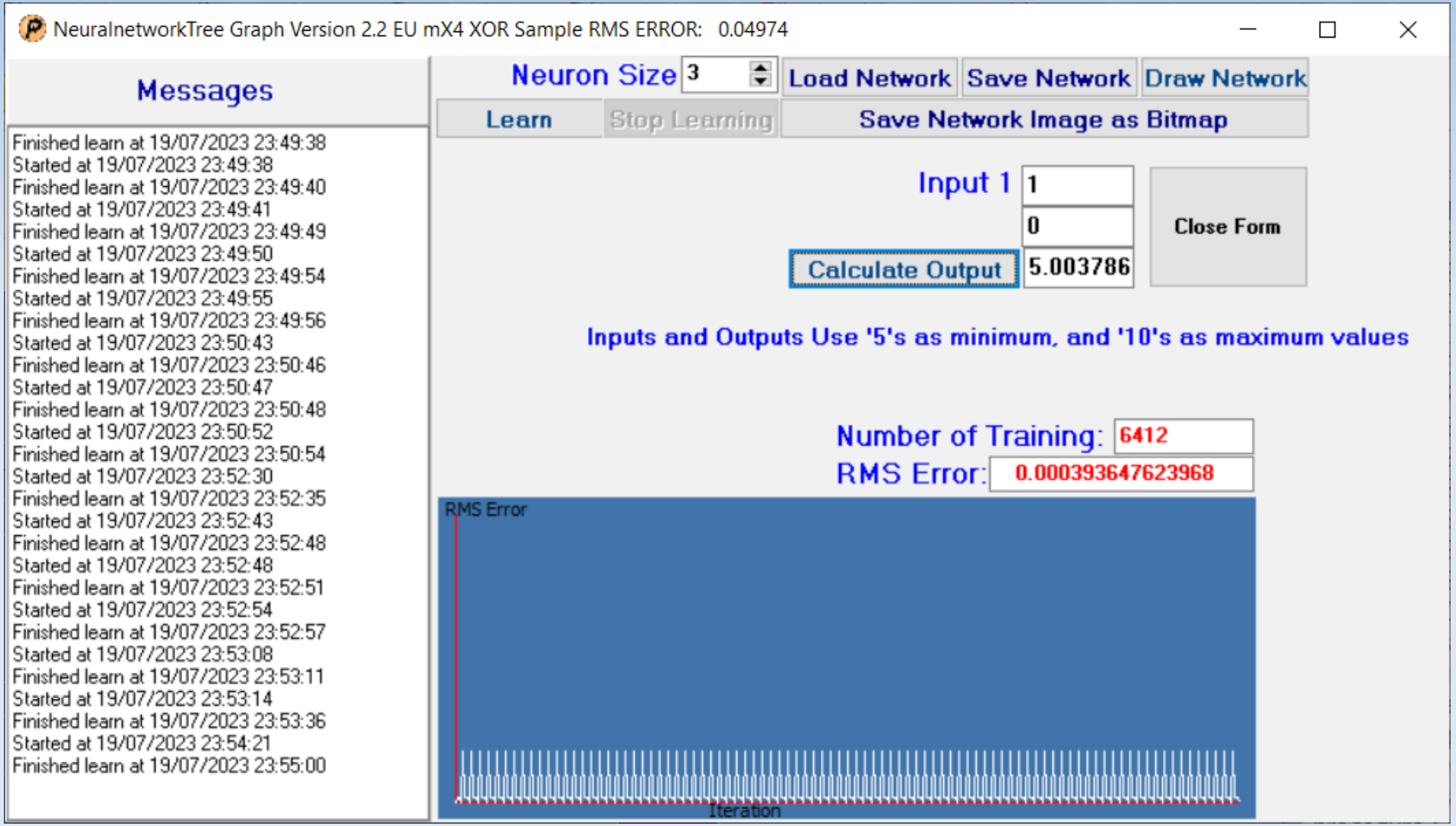
EKON 27

Graph Control



EKON 27

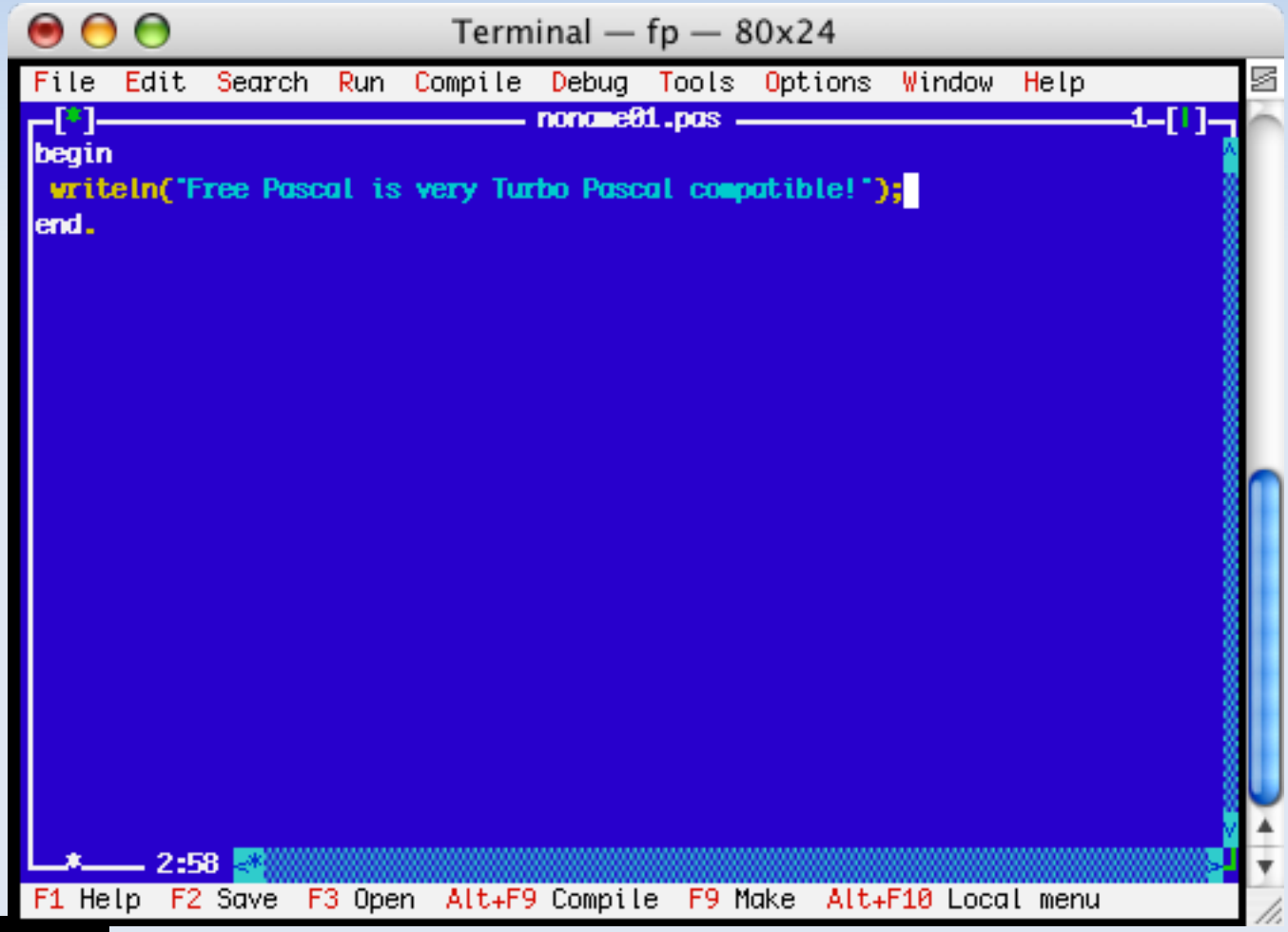
Flow & Graph Control II



Demo: 1234_NeuralNetwork2_XOR_sampleEU_EKON27.pas

ML Package

- IDE for Console or Terminal



EKON 27

MLP

- The neural-api or CAI API (Conscious Artificial Intelligence) is
- something like TensorFlow for Pascal and is platform-independent open source
- library for artificial intelligence or machine learning in the field of
- speech recognition, image classification, OpenCL, big data, data science, sentiment-analysis
- and computer vision2 with more or less SVG.

https://github.com/maxkleiner/neural-api/

Using MLP

- To be able to run this example, you'll need to load an already trained neural network file and then select the image you intend to classify.
- *TRAINPATH = '.\model\ClassifyCNNResize40_84.nn'*
- CAI stores both architecture and weights into the same *.nn file!
- Dropout is a simple and powerful regularization technique for neural networks and deep learning models.
- Download the package:

https://github.com/maxkleiner/neural-api/blob/master/examples/SimpleImageClassifier/MachineLearningPackage.zip

- Just unpack the zip and start exe

Let's compile

- As the name implies, it is a CNN-model. A Convolutional1 Neural Network (CNN) is a type of deep learning algorithm that is particularly for image recognition and object-detection tasks. It is made up of multiple layers, including convolution layers, pooling layers, and fully connected layers.
- Const PICPATH = '.\data\';
- TRAINPATH = '.\model\ClassifyCNNModel_70.nn';
- The main procedure to classify incoming images loads the model, decides dropout or not (later more) and creates input- and output-volumes with a shape of 32;32;3 or a 32x32x3 volume:
- Demo: *1135_classify_cifar10images1_5.pas*

The Main

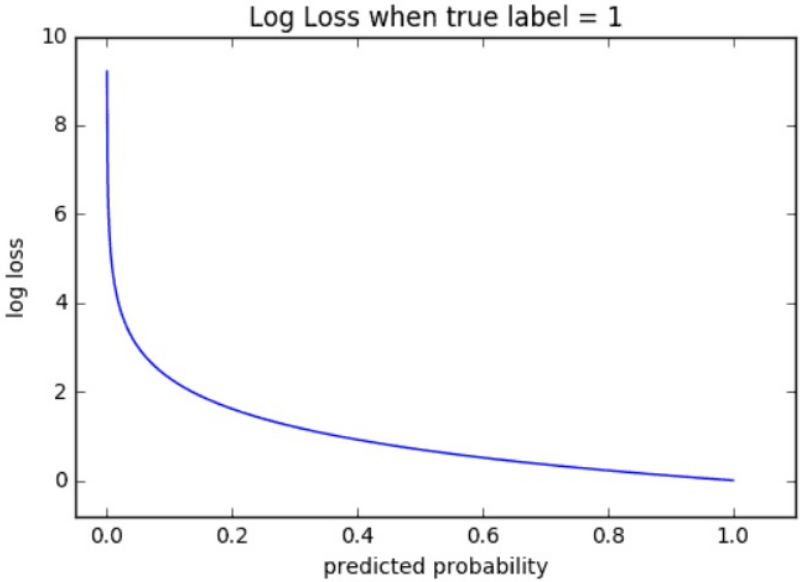
```
begin
  NN:= THistoricalNets.create; //TNNet.Create();
  NN.LoadFromFile(TRAINPATH);
  label2.caption:= 'load: '+TRAINPATH;

  if chkboxdrop.checked then
    NN.EnableDropouts(true) else
      NN.EnableDropouts(false);
  pInput:= TNNetVolume.Create0(32, 32, 3, 1);
  pOutPut:= TNNetVolume.Create0(10, 1, 1, 1);

  LoadPictureIntoVolume(image1.picture, pinput);
  pInput.RgbImgToNeuronalInput(csEncodeRGB);
  NN.Compute65(pInput,0);
  NN.GetOutput(pOutPut);
  writeln('result get class type: '+itoa(pOutPut.GetClass()));
```

Loading 10K images from file "C:\Program Files\Streaming\maxbox4\maxbox47590\maxbox4\cifar-10-batches-bin\test_batch.bin" ...
GLOBAL MIN MAX -2 1.984375
Testbatch size: 48000000
dbug volumescount: 10000
Ver: 4.7.6.50 (476). Workdir: C:\Program Files\Streaming\maxbox4\maxbox47590\maxbox4\resized\MLPackage
Testbatch score: Rate:0.8000, Loss:-0.0639, ErrorSum:467.7171

EKON 27

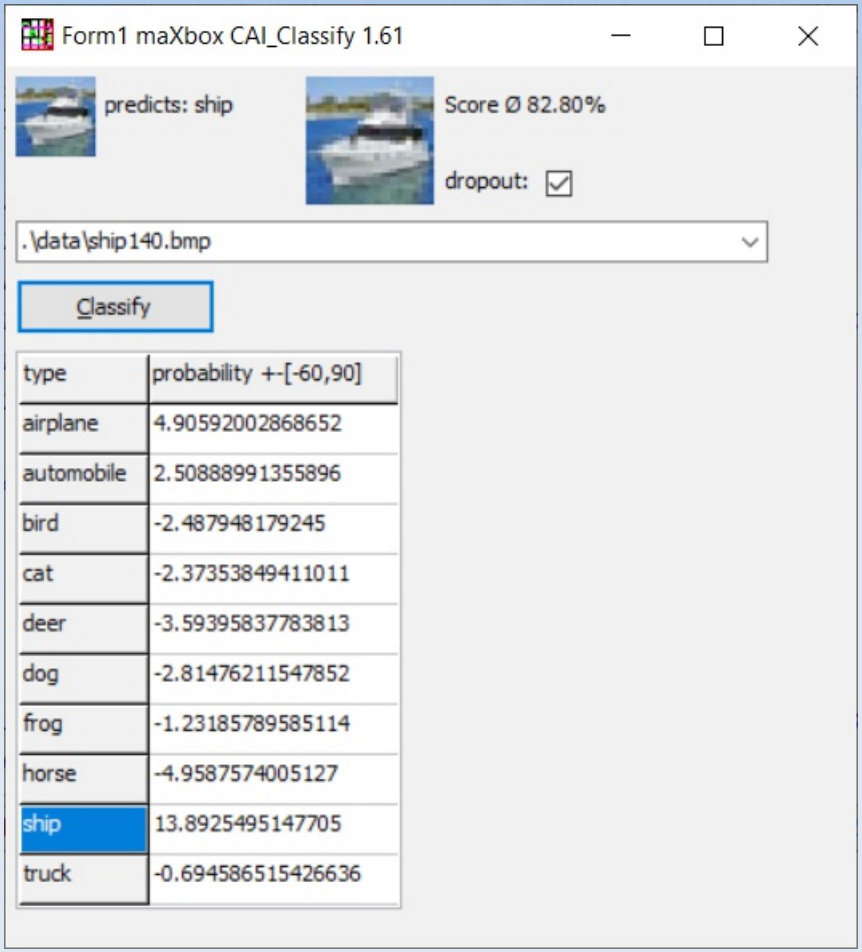
Pretrained

- The *.nn file in TRAINPATH serves as a pre-trained file (FAvgWeight) to classify/predict images we trained on. Also the CIFAR-10 with experiments/testcnnalgo/testcnnalgo.lpr and a number of CIFAR-10 classification examples are available on /experiments.
-
- Imagine the accuracy goes up and the loss-function (error-rate) goes down. The loss function is the bread & butter of modern machine learning; it takes your algorithm from theoretical to practical and transforms matrix multiplication into deep learning.



EKON 27

Electron Web App



- Unit classify_cifar10_images2lazTutor42_1_61;

Form Create

```
procedure TForm1FormCreate(Sender: TObject);
var k,t: integer;
  items: TStringList;
begin
  items:= TStringList.create;
  for k:= 0 to 9 do
    StringGrid1.Cells[0, k+1]:= cs10Labels[k];

  //FindAllFiles(ComboBox1.Items, 'csdata');
  FindFiles(exepath+'data', '*.bmp',items);
  writeln(items.text);
  for t:= 1 to items.count-1 do
     ComboBox1.Items.add(items[t]);
  if ComboBox1.Items.Count > 0 then begin
    ComboBox1.text:= ComboBox1.Items[0];
    if FileExists(ComboBox1.text) then begin
      Image1.Picture.LoadFromFile(ComboBox1.text);
      Image2.Picture.LoadFromFile(ComboBox1.text);
      label1.Caption:= extractfilename(ComboBox1.text);
    end;
  end;
end;
```


Performance

- Loss functions are different based on a problem statement to which deep learning is being applied. The cost function is another term used interchangeably for the loss function, but it holds a more different meaning.
- A loss function is for a single training example, while a cost function is an average loss over a complete train dataset.
- train neural network start..:15000 iterators, train neural network finished:
- RMS ERROR: 0.004679808474835: evaluate neural network..
- i:0 j:0 0.095534542493845
- i:0 j:1 0.931946667637693
- i:1 j:0 0.914564807733864
- i:1 j:1 0.005510785410457

debug inf nncount:3 mX4 executed: 14/08/2023 09:33:01 Runtime: 0:0:6.1 Memload: 54% use

EKON 27


Web Platforms

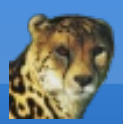
- As a Jupyter Notebook:
- https://github.com/maxkleiner/maXbox/blob/master/EKON24_SimpleImageClassificationCPU.ipynb
- and the same in Colab.research:
- https://colab.research.google.com/github/maxkleiner/maXbox/blob/master/EKON24_SimpleImageClassificationCPU.ipynb

The whole package with app, script, tutorial, data and model:

https://github.com/maxkleiner/neural-api/blob/master/examples/SimpleImageClassifier/MachineLearningPackage.zip

https://github.com/maxkleiner/maXbox/blob/master/objectdetector3.ipynb

EKON 27




Conclusion

- Only CNN is available but diff. structures possible, implement some of these well known architectures with CAI:

- For example Yann LeCun LeNet-5:
- NN := TNNet.Create();
- NN.AddLayer(TNNetInput.Create(28, 28, 1));
  NN.AddLayer(TNNetConvolution.Create(6, 5, 0, 1));
  NN.AddLayer(TNNetMaxPool.Create(2));
  NN.AddLayer(TNNetConvolution.Create(16, 5, 0, 1));
  NN.AddLayer(TNNetMaxPool.Create(2));
  NN.AddLayer(TNNetFullConnect.Create(120));
  NN.AddLayer(TNNetFullConnect.Create(84));
  NN.AddLayer(TNNetFullConnectLinear.Create(10));
  NN.AddLayer(TNNetSoftMax.Create());

Method: Classification
Model: **CNN** + CIFAR-10
Metric: RMS

https://raw.githack.com/breitsch2/maXbox4/master/assets/graph3.html

Thanks for coming!

Materials:

http://www.softwareschule.ch/download/maxbox_starter105.pdf

https://github.com/breitsch2/maXbox4/tree/master/assets