MLP Calculator

Practical User Manual

Comprehensive Operating Guide

English

Wolfgang Boehme

wolfgang.boehme@economic-modeling.de

www.economic-modeling.de

25 October 2026

The documentation on this website was developed with the support of the AI Claude (Anthropic).

Table of Contents

1 Quick Start – First Steps **2**
- 1.1 What You Need . . . 2
- 1.2 Minimal Workflow (5 Minutes) . . . 2

2 Preparing and Loading Data **3**
- 2.1 Preparing the CSV Format . . . 3
  - 2.1.1 Correct Structure . . . 3
  - 2.1.2 Important Points . . . 3
- 2.2 Loading CSV into the Calculator . . . 3
  - 2.2.1 Step 2: Choose Options . . . 3
  - 2.2.2 Step 3: Upload CSV . . . 4
  - 2.2.3 Step 4: Verification . . . 4
- 2.3 Testing Example Datasets . . . 4
  - 2.3.1 XOR Problem . . . 4
  - 2.3.2 Simple Regression . . . 5
  - 2.3.3 Circle Classification . . . 5

3 Configuring and Creating a Network **6**
- 3.1 Choosing the Architecture . . . 6
  - 3.1.1 Setting Parameters . . . 6
- 3.2 Creating the Network . . . 7

4 Running Training **8**
- 4.1 Starting Training . . . 8
  - 4.1.1 Epochs . . . 8
  - 4.1.2 Steps . . . 8
- 4.2 Monitoring Training . . . 8
  - 4.2.1 Interpreting the Loss Curve . . . 8
  - 4.2.2 When is it Good Enough? . . . 8
- 4.3 Training Complete . . . 9
  - 4.3.1 Automatic Backtesting . . . 9

5 Making Predictions **10**
- 5.1 Manual Input . . . 10
- 5.2 Batch Predictions . . . 10
  - 5.2.1 Metrics . . . 10
  - 5.2.2 Understanding Metrics . . . 10

6 Analysing and Exporting Results **11**
- 6.1 Visualisations . . . 11
- 6.2 Exporting Data . . . 11
  - 6.2.1 Model Data . . . 11
  - 6.2.2 Training & Predictions . . . 11
  - 6.2.3 Visualisations . . . 11
- 6.3 Working Further in Excel . . . 11

7 Common Problems and Solutions **12**
- 7.1 When Loading Data . . . 12
  - 7.1.1 “No valid data found” . . . 12
  - 7.1.2 “Input neurons do not match” . . . 12
- 7.2 During Training . . . 12

7.2.1 Loss does not decrease . . . 12
7.2.2 Loss jumps wildly . . . 12
7.2.3 Loss stagnates at high value . . . 12
7.2.4 “NaN” in training . . . 12
7.3 With Predictions . . . 12
7.3.1 Predictions always the same . . . 12
7.3.2 Predictions completely wrong . . . 12
7.3.3 Test-set error much higher than training . . . 12

8 Best Practices **13**
8.1 Network Configuration Rules . . . 13

9 Complete Example Workflow **14**
9.1 Scenario: Currency Rate Prediction . . . 14
9.1.1 1. Prepare Data (in Excel) . . . 14
9.1.2 2. Open the Calculator . . . 14
9.1.3 3. Tab “Network” – Configuration . . . 14
9.1.4 4. Tab “Data” – Loading . . . 14
9.1.5 5. Tab “Training” – Train . . . 14
9.1.6 6. Tab “Prediction” – View Results . . . 15
9.1.7 7. Manual Prediction . . . 15
9.1.8 8. Export and Document . . . 15
9.1.9 9. Excel Analysis . . . 15
9.1.10 10. Summary . . . 15

10 Summary **16**
10.1 Workflow in Brief . . . 16
10.2 Most Important Points . . . 16

1 Quick Start – First Steps

1.1 What You Need

- A CSV file with your data
- Format: first columns = inputs, last column = output
- Example: 5 exchange rates (inputs) → 1 predicted value (output)

1.2 Minimal Workflow (5 Minutes)

1. **Tab "Data"** → Upload CSV
2. **Tab "Network"** → Click "Create Network"
3. **Tab "Training"** → Click "Start Training"
4. Wait until training is complete
5. **Tab "Prediction"** → Enter values → "Calculate Prediction"

Success

Done! You have trained your first neural network!

2 Preparing and Loading Data

2.1 Preparing the CSV Format

2.1.1 Correct Structure

```
Input1,Input2,Input3,Input4,Input5,Output
12503.93,13341.24,1745.45,15425.45,14183.48,11439.91
12465.50,13356.22,1747.25,15466.59,14154.82,11438.08
12432.37,13310.80,1748.69,15504.41,14115.41,11422.34
```

Listing 1: Example CSV file

2.1.2 Important Points

Information

Correct:

- Header row (optional, but recommended)
- Commas as delimiters
- Full stops for decimals (not commas!)
- No empty rows
- No special characters
- Last column = output (what is to be predicted)

Error

Incorrect:

- Date column at the start (remove it!)
- Empty columns at the end
- Text in data rows
- European number format (12.503,93) instead of (12503.93)

2.2 Loading CSV into the Calculator

2.2.1 Step 2: Choose Options

Option A – Normalise data: Activate when:

- Your values differ greatly in magnitude
- Example: Input1=12000, Input2=1.5 (factor of 8000!)
- Currency data: YES, activate! ✓

Do not activate when:

- All values between 0 and 1
- Already normalised data

Option B – Train/Test Split: Activate (recommended!):

- Standard: 80% training, 20% test
- System automatically tests on unseen data
- Honest evaluation of performance

Choosing the ratio:

- 50–100 data points: 85/15 or 90/10
- 100–500 data points: 80/20 ✓(standard)
- 500+ data points: 70/30

2.2.2 Step 3: Upload CSV

1. Click: "Upload CSV file"
2. Select file
3. Wait until "CSV loaded" appears

2.2.3 Step 4: Verification

The system automatically displays:

- Number of samples
- Train/test split
- Data preview (first rows)
- Statistics (min, max, mean)

Success

Example output:
Train/Test Split active:
Total: 252 samples
Training: 201 samples (80%)
Test: 51 samples (20%)
Normalisation: Data normalised to [0, 1]

2.3 Testing Example Datasets

Before using your own data, test with the examples:

2.3.1 XOR Problem

- 2 inputs, 1 output; 4 samples
- Tests whether the network learns nonlinearly
- Recommended for the first test!

Usage:

1. Tab "Network" → Input neurons: 2
2. Tab "Data" → Click button "XOR Problem"
3. Tab "Training" → Start training
4. Should reach loss < 0.01

2.3.2 Simple Regression

1 input, 1 output, 50 samples. Good for learning the tool.

2.3.3 Circle Classification

2 inputs, 1 output, 200 samples. Good for testing train/test split.

3 Configuring and Creating a Network

3.1 Choosing the Architecture

Open tab "Network".

3.1.1 Setting Parameters

Input neurons:

- = number of your input columns
- Example currencies: 5 (USD, EUR, CNY, GBP, CHF)
- MUST match the CSV!

Hidden Layer 1:

- Size: 6–20 neurons
- Rule of thumb: 2× to 4× input neurons
- Example: Input=5 → Hidden1=12 ✓
- Activation: ReLU (standard, works well)

Hidden Layer 2 (optional):

- Size: half of Hidden Layer 1
- Example: Hidden1=12 → Hidden2=6 ✓
- Tip: add for complex problems

Output:

- = 1 (for predicting a value)
- Activation: ReLU for regression, Sigmoid for binary classification

Learning rate:

- Standard: 0.001 ✓(usually works well)
- Too small (0.0001): learns very slowly
- Too large (0.01): can become unstable

Information

Example configuration (currencies):
Input: 5 (USD, EUR, CNY, GBP, CHF)
Hidden Layer 1: 12, Activation: ReLU
Hidden Layer 2: 6, Activation: ReLU
Output: 1, Activation: ReLU
Learning rate: 0.001

3.2 Creating the Network

1. Check all parameters
2. Click “Create Network”
3. System shows: “Network created with X parameters”
4. Open tab “Visualisation” to verify the network graph

Error

Common error:
Input neurons do not match CSV
→ Error message during training
→ Solution: count columns in CSV!

4 Running Training

4.1 Starting Training

Open tab "Training".

4.1.1 Epochs

- = number of passes through all data
- Standard: 1 000–2 000
- Small data (<100): 3 000–5 000
- Large data (>1 000): 500–1 000
- You can press Stop at any time!

4.1.2 Steps

Step 1: Set epochs (recommendation: 2 000 for currency data)

Step 2: Click "Start Training"

Step 3: Monitor the display:

- Current epoch
- Training loss (should decrease!)
- Best loss
- Elapsed time
- Progress bar

4.2 Monitoring Training

4.2.1 Interpreting the Loss Curve

DESIRED: Curve falls steadily
OK: Curve falls then flattens
PROBLEM: Curve rises or jumps wildly

4.2.2 When is it Good Enough?

Regression (predicting numbers):

- ✓ Loss < 0.01: Excellent!
- ✓ Loss < 0.05: Good
- ○ Loss < 0.1: OK, could be better
- × Loss > 0.5: Poor, train longer

Classification (0 or 1):

- ✓ Loss < 0.1: Excellent!
- ✓ Loss < 0.3: Good
- ○ Loss < 0.5: OK
- × Loss > 0.7: Poor

4.3 Training Complete

Success

Training complete!
Total time: 25 s
Final loss: 0.002456
Starting automatic backtesting with test data...
Test-set backtesting complete

4.3.1 Automatic Backtesting

- Only when train/test split was active
- System tests with 20% unseen data
- Shows real performance!
- Tab "Prediction" for results

5 Making Predictions

5.1 Manual Input

Open tab "Prediction".

Warning

IMPORTANT: ALWAYS enter original values!
Example: 12503.93, 13341.24, 1745.45, 15425.45, 14183.48
NOT: normalised values (0.4782, 0.1881, ...)
The system normalises automatically internally!

Success

Prediction result:
Your input (original values): [12503.93, 13341.24, ...]
Passed to network (normalised): [0.4782, 0.1881, ...]
Prediction (original scale): 11439.45

5.2 Batch Predictions

For many predictions at once – upload a CSV with inputs only (no output column) or with output column for backtesting.

5.2.1 Metrics

Success

Backtesting metrics:
MAE: 0.0204 → Average error
RMSE: 0.0294 → Root mean square error
MAPE: 0.18% → Error in percent
Test samples: 51

5.2.2 Understanding Metrics

- MAE near 0: Excellent!
- MAPE $< 1\%$: Excellent!
- MAPE 1–5%: Good
- MAPE $> 10\%$: Needs improvement

6 Analysing and Exporting Results

6.1 Visualisations

Open tab "Visualisation".

- **Network graph:** Architecture, all layers and connections; export as PNG
- **Loss curve:** X-axis: epochs, Y-axis: loss (should decrease!); export as PNG
- **Weight distribution:** Histogram, should be bell-shaped; extremes = problem
- **Activation values:** Table of all neurons in the last prediction

6.2 Exporting Data

Open tab "Export".

6.2.1 Model Data

- **Weights (CSV):** All weights and biases, for documentation
- **Architecture (CSV):** All parameters, network structure
- **Activation values (CSV):** All neuron activations, for debugging

6.2.2 Training & Predictions

- **Training history (CSV):** Epoch, loss, time – for charts in Excel
- **Batch predictions (CSV):** All predictions in original scale
- **Test-set backtesting (CSV):** Unseen data, real performance, MAE, RMSE, MAPE

6.2.3 Visualisations

Network graph (PNG), loss curve (PNG), weight distribution (PNG)

6.3 Working Further in Excel

After CSV export:

- Create chart from training history: Epoch vs. Loss
- Scatter plot: Prediction vs. Actual
- Calculate metrics: `=AVERAGE(ABS(errors))`, `=SQRT(AVERAGE(errors^2))`

7 Common Problems and Solutions

7.1 When Loading Data

7.1.1 "No valid data found"

Cause: Wrong CSV format
Solution: Header row? Commas as delimiters? Full stops for decimals? No empty rows?

7.1.2 "Input neurons do not match"

Cause: Wrong number of input neurons set
Solution: Count columns in CSV (excluding output!), adjust in tab "Network", recreate network.

7.2 During Training

7.2.1 Loss does not decrease

Increase learning rate ($0.001 \rightarrow 0.005$), more epochs, larger network.

7.2.2 Loss jumps wildly

Reduce learning rate ($0.001 \rightarrow 0.0005$), restart training.

7.2.3 Loss stagnates at high value

More hidden neurons, add second hidden layer, check activation (ReLU for regression!).

7.2.4 "NaN" in training

Strongly reduce learning rate (0.0001), recreate network, restart training.

7.3 With Predictions

7.3.1 Predictions always the same

Train longer, choose better architecture, check data.

7.3.2 Predictions completely wrong

Always enter original values (NOT normalised!), reload CSV with normalisation, retrain.

7.3.3 Test-set error much higher than training

Overfitting: fewer epochs, smaller network, more training data.

8 Best Practices

Success

Recommended:

- ALWAYS use train/test split (80/20)
- ALWAYS normalise data when magnitudes differ greatly
- Check CSV in Excel before upload
- At least 50–100 samples for good results
- More data = better results

Error

Avoid:

- No date columns in CSV
- No empty rows
- No text columns (except header)
- Too little data (<20 samples)

8.1 Network Configuration Rules

- **Hidden1:** 2× to 4× input neurons
- **Hidden2:** Half of Hidden1
- **Learning rate start:** 0.001; too slow → 0.005; unstable → 0.0001
- **Epochs start:** 2 000; small data: 3 000–5 000
- **Activations:** Regression: ReLU → ReLU → ReLU; Classification: ReLU → ReLU → Sigmoid

9 Complete Example Workflow

9.1 Scenario: Currency Rate Prediction

Goal: Predict a currency value from 5 other currencies

9.1.1 1. Prepare Data (in Excel)

Create CSV with 252 rows, 5 input columns + 1 output column, no empty rows, full stops for decimals. Save as: `currency_data_original.csv`

9.1.2 2. Open the Calculator

Browser: open `mlp_calculator_v18_final.html`

9.1.3 3. Tab "Network" – Configuration

- Input neurons: 5
- Hidden Layer 1: 12 neurons, Activation: ReLU
- Hidden Layer 2: 6 neurons, Activation: ReLU
- Output neurons: 1, Activation: ReLU
- Learning rate: 0.001

Click: "Create Network"

Success

Result: "Network created with 157 parameters"

9.1.4 4. Tab "Data" – Loading

Activate normalisation and train/test split (80%). Upload `currency_data_original.csv`.

Success

System shows: Total 252 samples, Training 201 (80%), Test 51 (20%)
Normalisation: Data normalised to [0, 1]

9.1.5 5. Tab "Training" – Train

Set epochs: 2 000. Click "Start Training". After ∼2 000 epochs: done!

Success

Training complete! Time: 28 s
Final loss: 0.002891
Test-set backtesting: MAE=0.0198, MAPE=0.17%

9.1.6 6. Tab "Prediction" – View Results

Success

Backtesting metrics:
MAE: 0.0198 RMSE: 0.0287 MAPE: 0.17% Test samples: 51

RATING: MAPE 0.17% = Excellent! ✓

9.1.7 7. Manual Prediction

Enter: Input 1: 12500, Input 2: 13350, Input 3: 1750, Input 4: 15450, Input 5: 14180
Click: "Calculate Prediction"

Success

Result (original scale): 11440.12

9.1.8 8. Export and Document

Export: test-set backtesting (CSV), training history (CSV), architecture (CSV), loss chart (PNG).

9.1.9 9. Excel Analysis

Open `test_set_backtesting.csv`. Create scatter plot: Prediction vs. Actual. Points very close to 45° line → model works very well!

Error analysis: Max absolute error: 8.23; Median error: 1.85 – all errors < 10 units on values around 11 440 → $< 0.1\%$ error. Excellent!

9.1.10 10. Summary

Success

DONE! Model results:
Architecture: 5→12→6→1, ReLU
Training: 201 samples, 2 000 epochs, loss: 0.0029
Test: 51 samples, MAE: 0.0198, MAPE: 0.17%
Rating: Excellent – ready for production! ✓

10 Summary

The MLP Calculator enables you to:

- ✓ Create neural networks without programming
- ✓ Professional training with train/test split
- ✓ Automatic backtesting with unseen data
- ✓ Export all results for Excel analysis
- ✓ Visualisations for presentations

10.1 Workflow in Brief

1. Data → 2. Network → 3. Training → 4. Test → 5. Export

10.2 Most Important Points

- ALWAYS train/test split (80/20)
- ALWAYS normalise data for large values
- ALWAYS enter original values for predictions
- Test-set performance is the true metric!
- MAPE $< 1\%$ = Excellent!

Good luck with your neural networks!