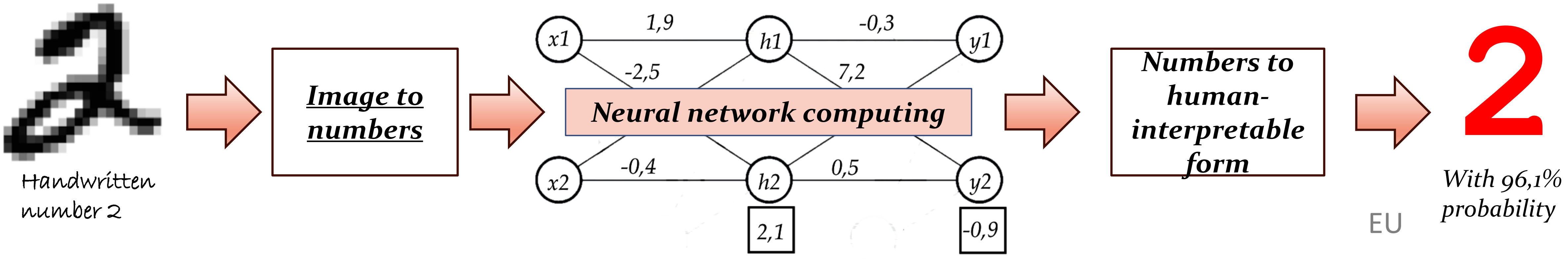
How neural network calculation works

Neural network, the heart and engine of AI

- A neural network is the basis for the operation of AI, a computational framework that derives output data from input data. In a neural network, all information is numerical, and all operations are mathematical operations.
- Neural networks are built and trained to perform specific artificial intelligence tasks.
- When the trained neural network is ready to perform the task, the AI application uses it as follows:
  1. The input information for AI, such as text, voice or image, is converted into numerical form and entered into the neural network.
  2. With the numerical input it receives, the neural network performs a series of calculations based on the numbers contained in the neural network, i.e. the calculation parameters. The result of the calculation is the neural network's output data in numerical form.
  3. Finally, the numerical result of the calculation performed by the neural network is converted into a human-interpretable form, such as text, sound or image. This is the AI's response to the input provided.

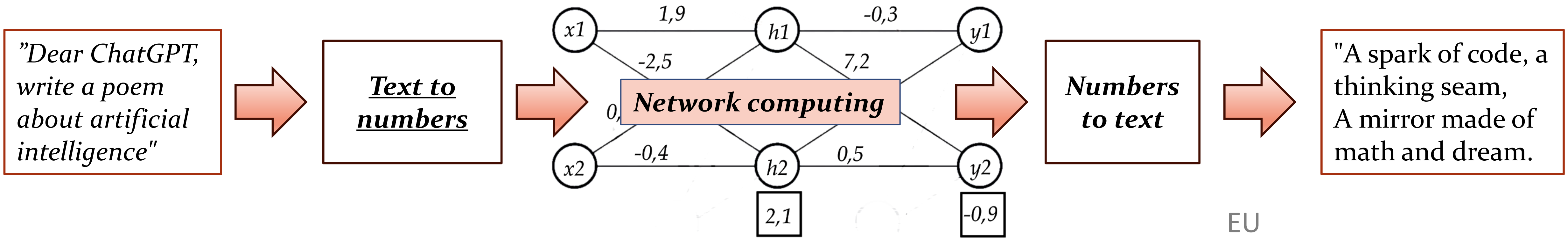


Examples of neural network tasks

Input data / prompt	**Task**
Photo of a person	Estimate age and gender
Audio file of birdsong	Estimate the bird's species
Laboratory results, X-ray picture	Make a diagnosis
Prompt text for image creation	Create an image

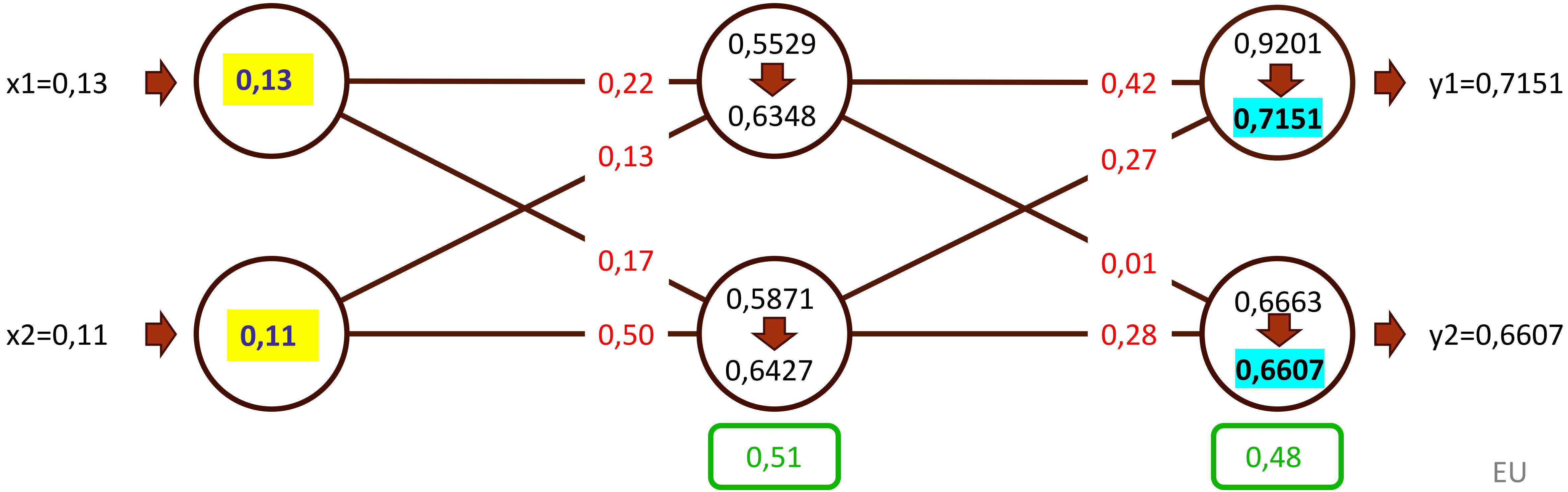
The tasks of the neural networks presented above are straightforward and easy to understand. The task of generative text-generating AI applications, on the other hand, is easily misunderstood, as the task of the neural network is to calculate the most likely continuation of the given prompt, not to provide the correct answer or a sensible comment on the input text:

Input data / prompt	**Task**
Prompt for a generative AI application	Create a likely continuum to the given prompt



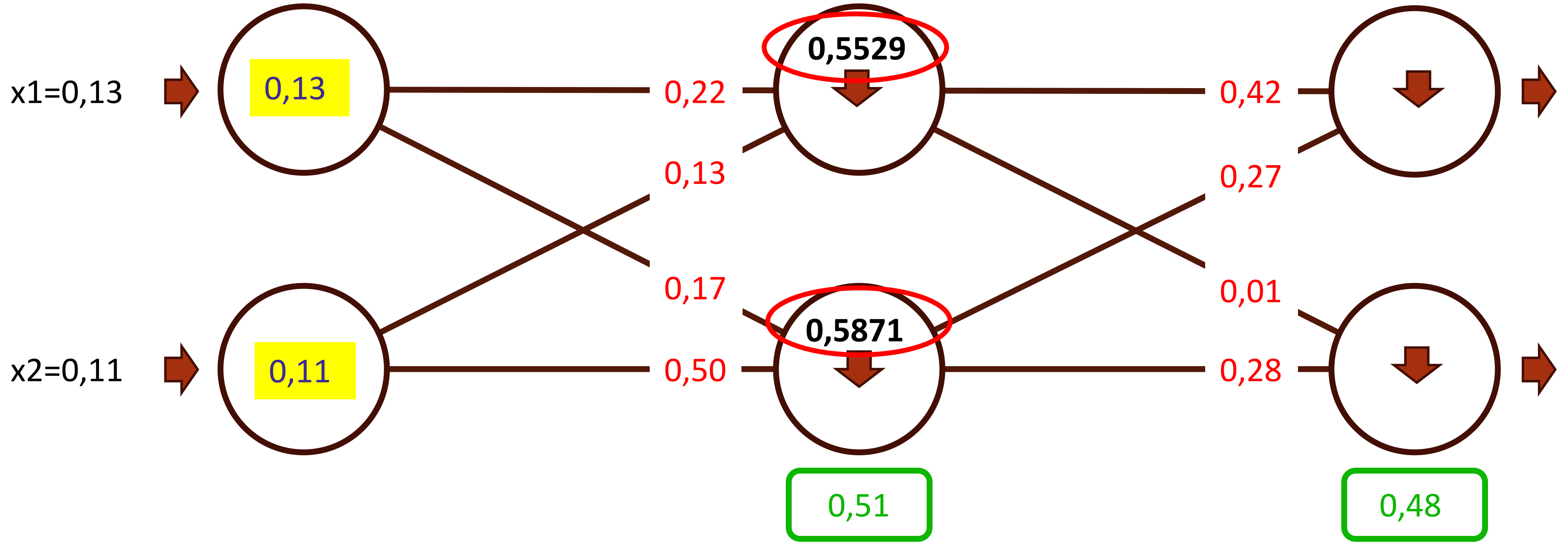
Neural network calculation 1/10, neural network structure

- An artificial neural network consists **of neurons (circles)** and **synapses** connecting them.
- The neural network calculation is based on **calculation parameters**, which are **the weights** attached to the synapses and **the bias values** attached to each layer of the network.
- The calculation proceeds from left to right from **the input values** (**x1** and **x2** in the example) until the **output values** are reached (**y1** and **y2** in the example).



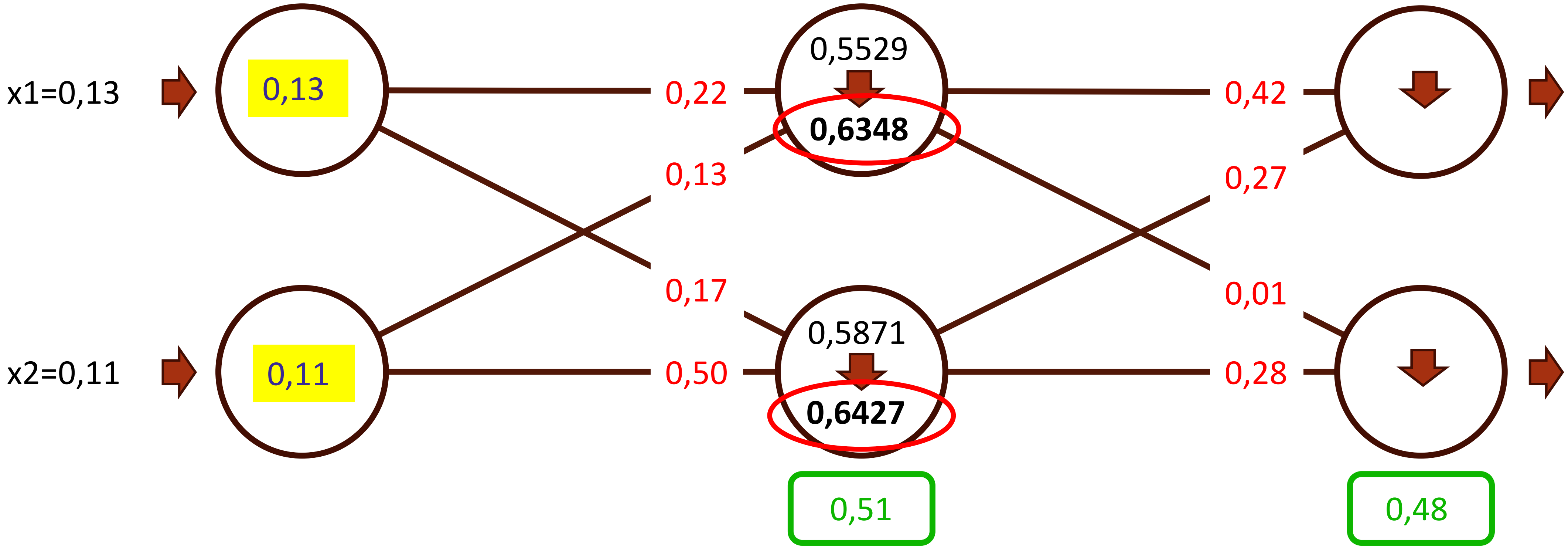
Neural network calculation 2/10, the input value of the neuron

- The input values of the neurons are obtained by multiplying the values of the preceding neurons by **the weights of the corresponding synapses**, adding up the products and the **bias value** attached to the neuron layer. The input values of the intermediate neurons in the example are thus calculated as follows:
  - Upper neuron: 0,13*0,22+0,11*0,13+0,51 = **0,5529**
  - Lower neuron: 0,13*0,17+0,11*0,50+0,51 = **0,5871**



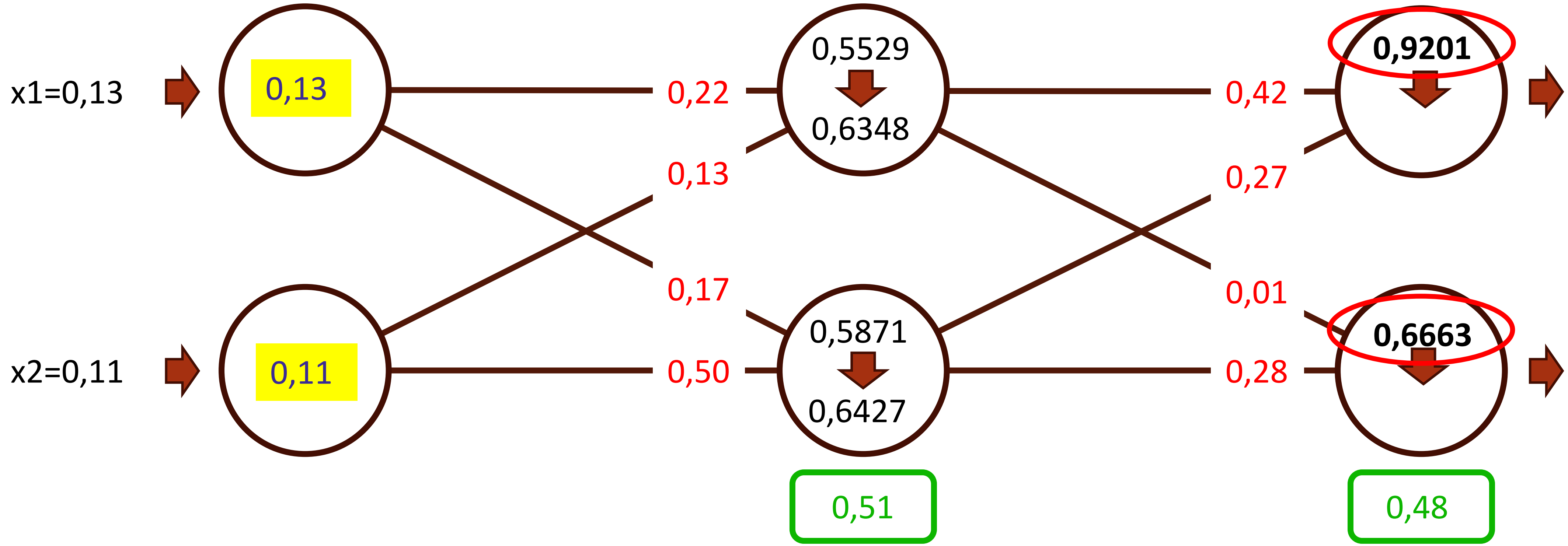
Neural network calculation 3/10, the output value of the neuron

- Before the value of the neuron is carried forward to the next layer, it is placed in a function called the **activation function**. Neural networks use different activation functions. In this example, the activation function is the **sigmoid function, i.e. 1/(1+e^⁻x),** so the output values of the intermediate neurons are:
  - Upper neuron: 1/(1+e^⁻ 0,5529) = **0,6348**
  - Lower neuron: 1/(1+e^⁻0,5871) = **0,6427**



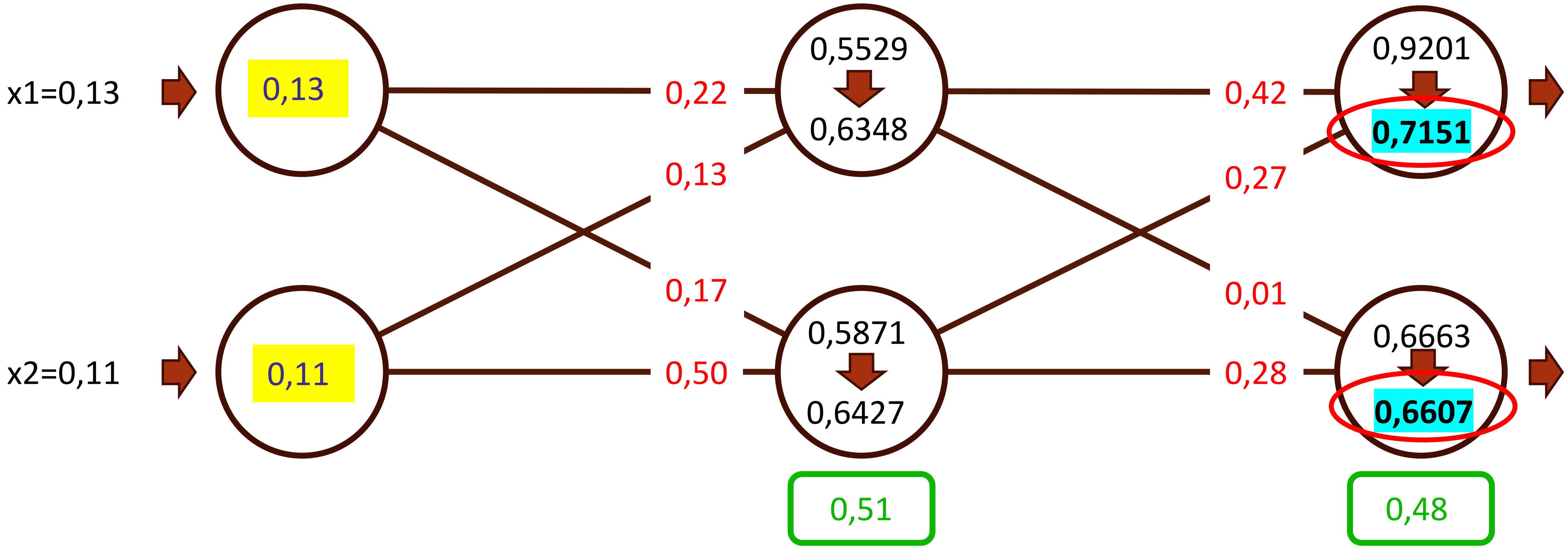
Neural network calculation 4/10, the calculation continues in the same way ...

- The output values of the neurons are further used to calculate the input values of the neurons in the next layer:
  - Upper neuron: 0,6348*0,42+0,6427*0,27+0,48 = **0,9201**
  - Lower neuron: 0,6348*0,01+0,6427*0,28+0,48 = **0,6663**



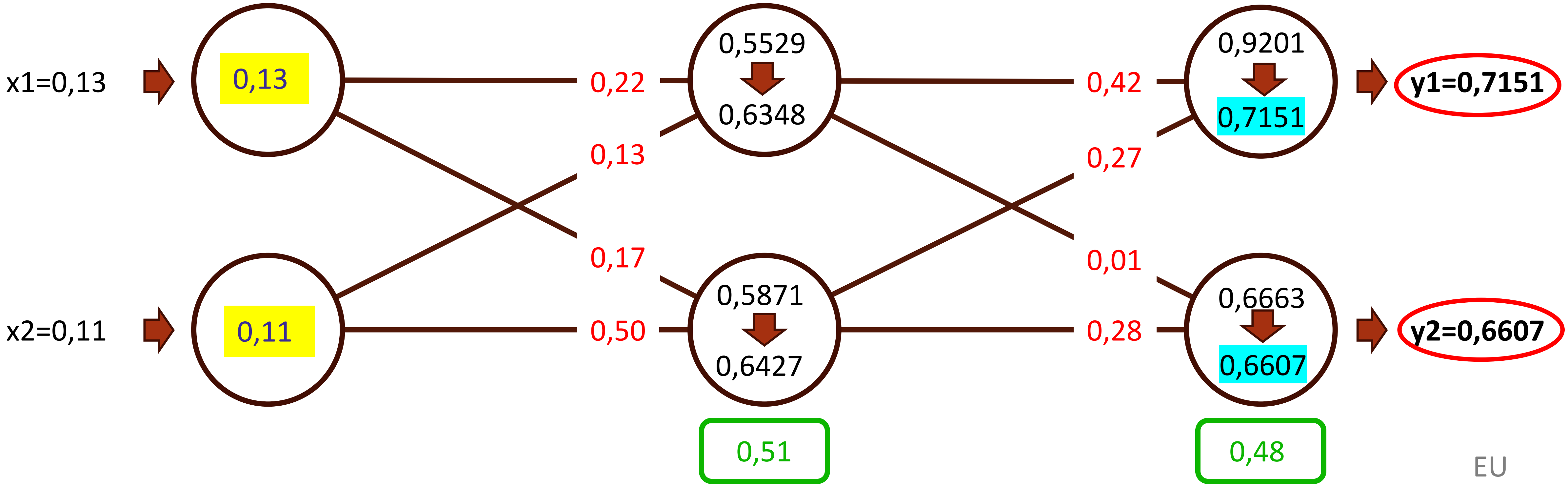
Neural network calculation 5/10, the calculation continues in the same way ...

- ... and the calculated input values of the neurons are again placed in the activation function, giving the output values of the neurons in this layer:
  - Upper neuron: 1/(1+e^- 0,9201) = **0,7151**
  - Lower neuron: 1/(1+e^-0,6663) = **0,6607**



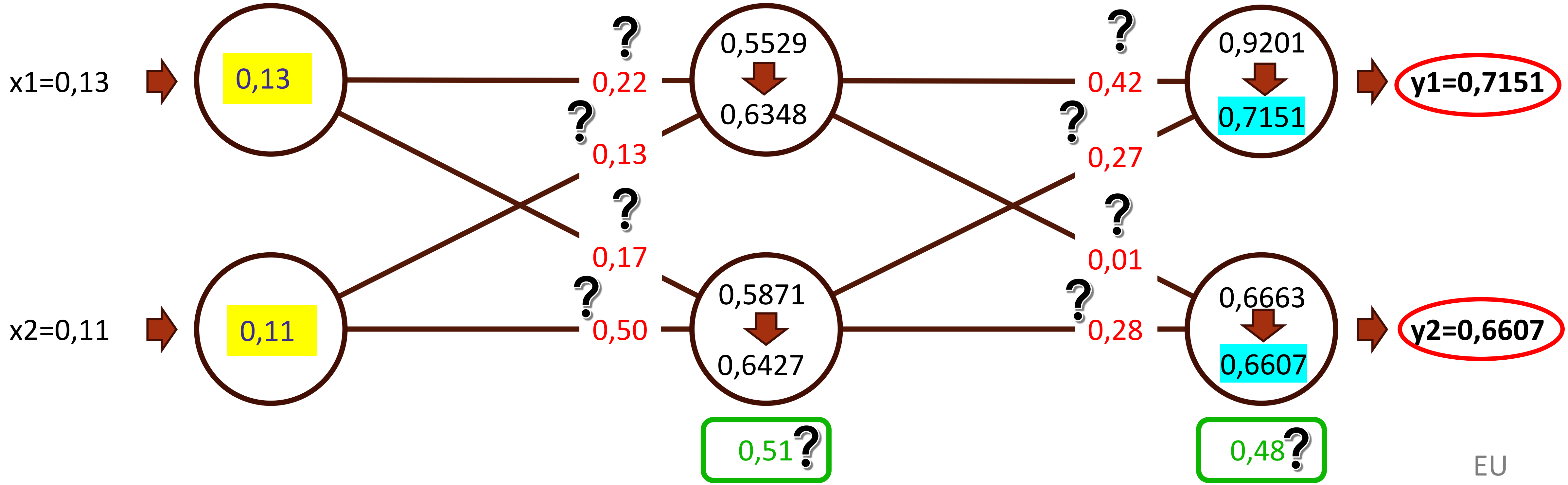
Neural network calculation 6/10, the result of the calculation

- The calculation continues in the same way throughout the neural network until the output values, in the example **y1** and **y2**, are obtained.
- So, we went through the entire neural network calculation from start to finish, **but something essential remained unexplained — what?**



Neural network calculation 7/10: How to obtain good values for the calculation parameters?

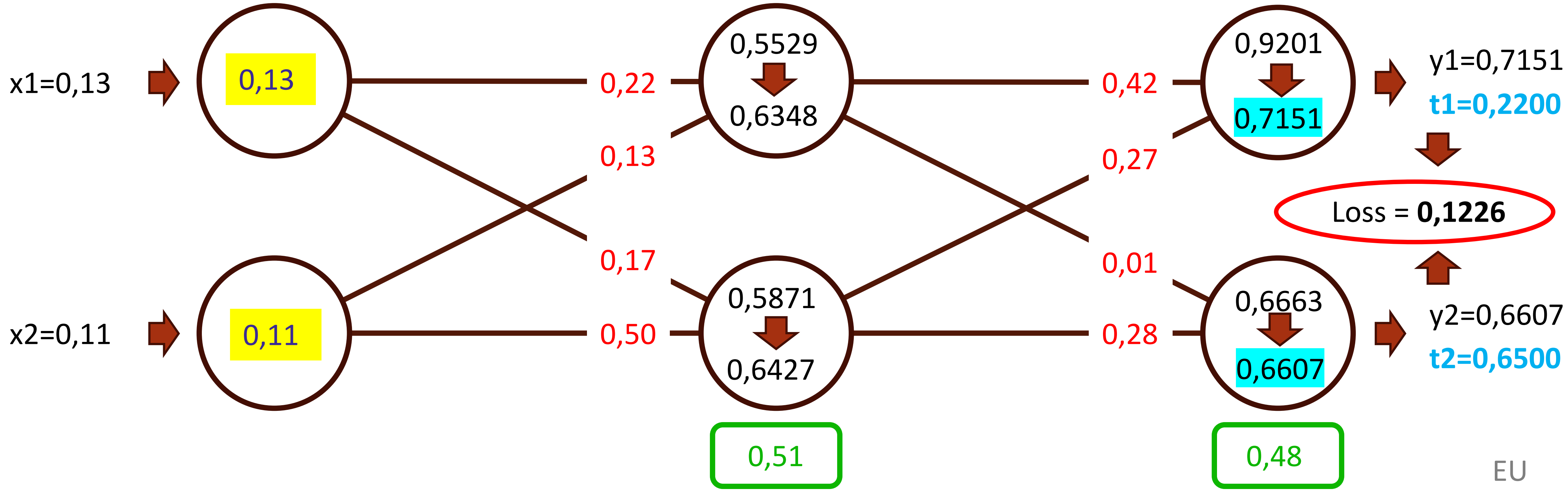
- As seen, the result of the calculation of the neural network is determined by the values of the calculation parameters. **So, how do we obtain calculation parameter values, weights and biases, which give good results?**
- **First, we need to determine "good result".** This is based on **training material**, which contains both the input data and the pre-known output data as desired target results. A good result differs as little as possible from the known target results.



Neural network calculation 8/10, measuring the success of the calculation

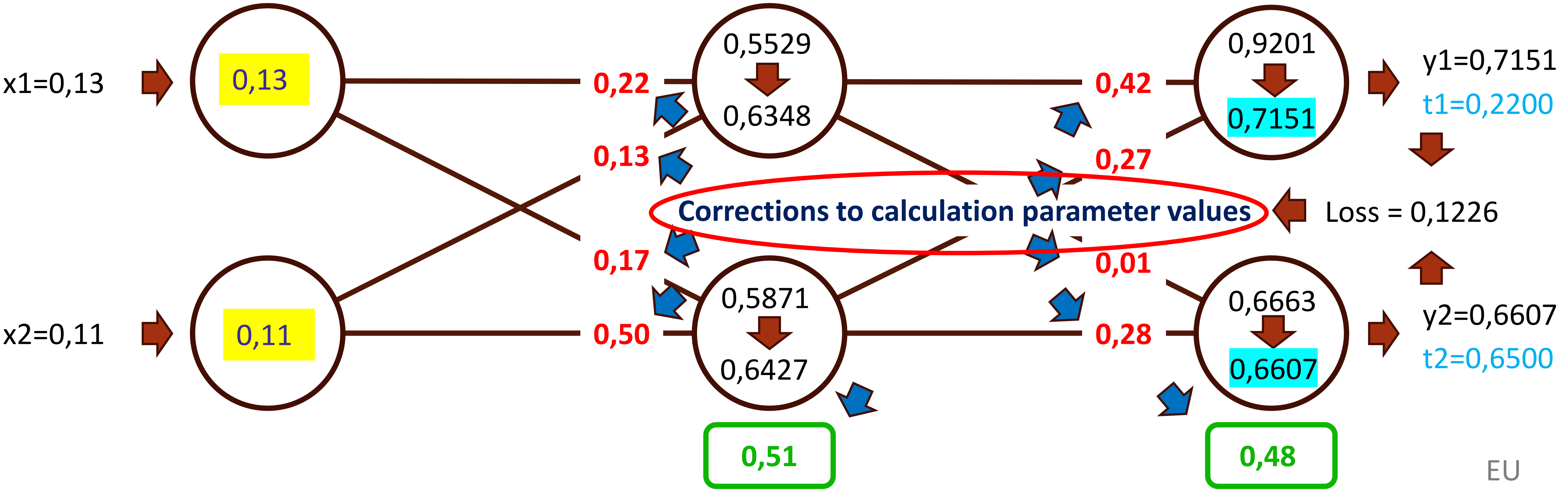
- Using the training material with both the input data and known target results, we can measure the success of the calculation by comparing the calculated results with the target results. For this, a so-called **loss function**, which describes the magnitude of the error, is determined. In this example, we use **the mean square error function** as the loss function. The value of the mean square error calculated from the **final results (y1 and y2)** and **the target results (t1 and t2)** in the example is:

Loss = $1/2((y1-t1)^2+(y2-t2)^2) = 1/2((0{,}7151-0{,}2200)^2+(0{,}6607-0{,}6500)^2)$ = **0,1226**



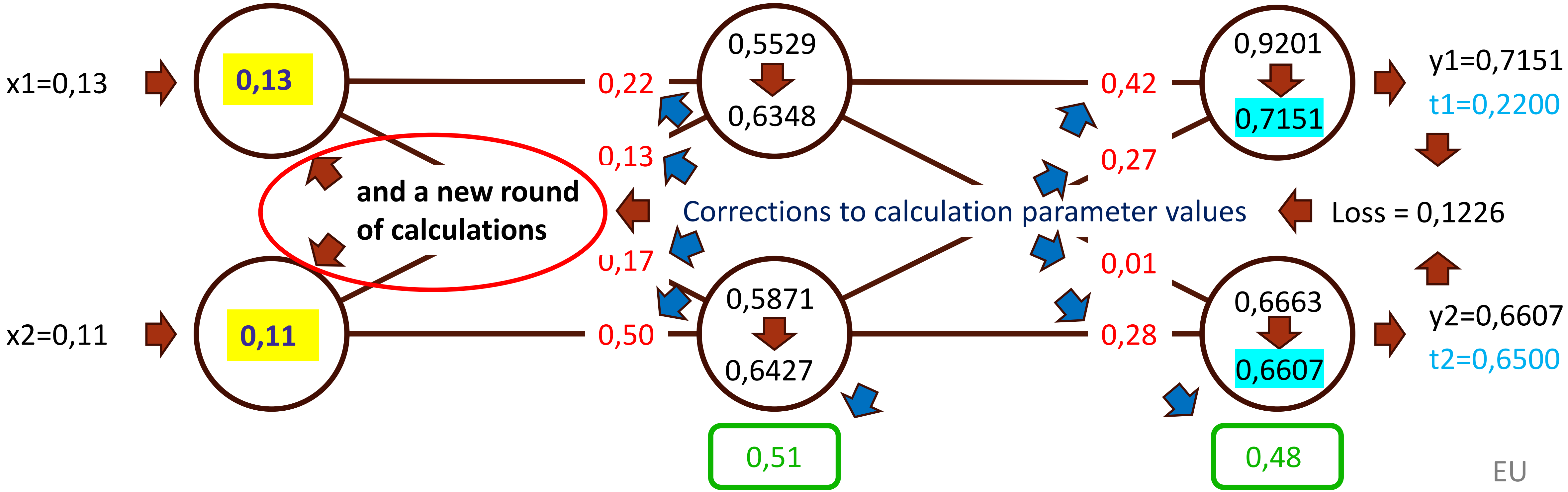
Neural network calculation 9/10, network training

- When we can measure the success of the calculation, we can **train the neural network to produce the best and most relevant results possible.**
- The training begins by studying in which direction the calculation parameter values of the neural network should be changed to reduce the value of the loss function, i.e. the error in the network calculation, and by changing the values by a small step in that direction.



Neural network calculation 10/10, network training

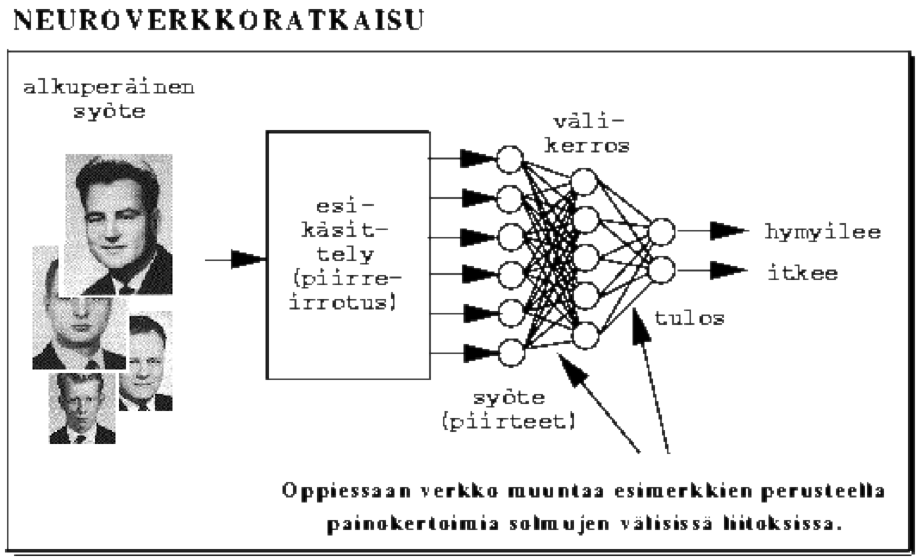
- When the values of the calculation parameters have been adjusted a small step in the right direction, **the value of the loss function is recalculated for the entire training material** or a randomly selected subset of it.
- The calculation is repeated in small increments until the value of the loss function is small enough.
- **As a result, the network can produce good results also from completely new and unknown material, which acts like the training material.**



Stages of the operation of a neural network

- **Defining the network's task and reviewing the prerequisites for implementation**: is there sufficiently extensive and high-quality data available for use in network training?
- **Network design:**
  - Select input and output data and specify how to convert input data into numerical form and numerical output data into human-interpretable form.
  - Define the structure of the network and select calculation formulas etc.
  - Select and prepare training and test materials.
- **Setting the input values of the calculation parameters:** Input values are typically given as random values, because there is no better basis for the Input values.
- **Training:** A large amount of training material, from which both input data and the desired target results are known, is passed through the neural network. The network is trained by adjusting its parameters to minimise the distance between its outputs and the target results.
- **Testing:** The neural network is tested with data from which both the input data and the desired target results are known, but which were not included in the training material.
- **Use:** As a result, the neural network can produce good results even from new and unknown source data, provided that it behaves like training material.

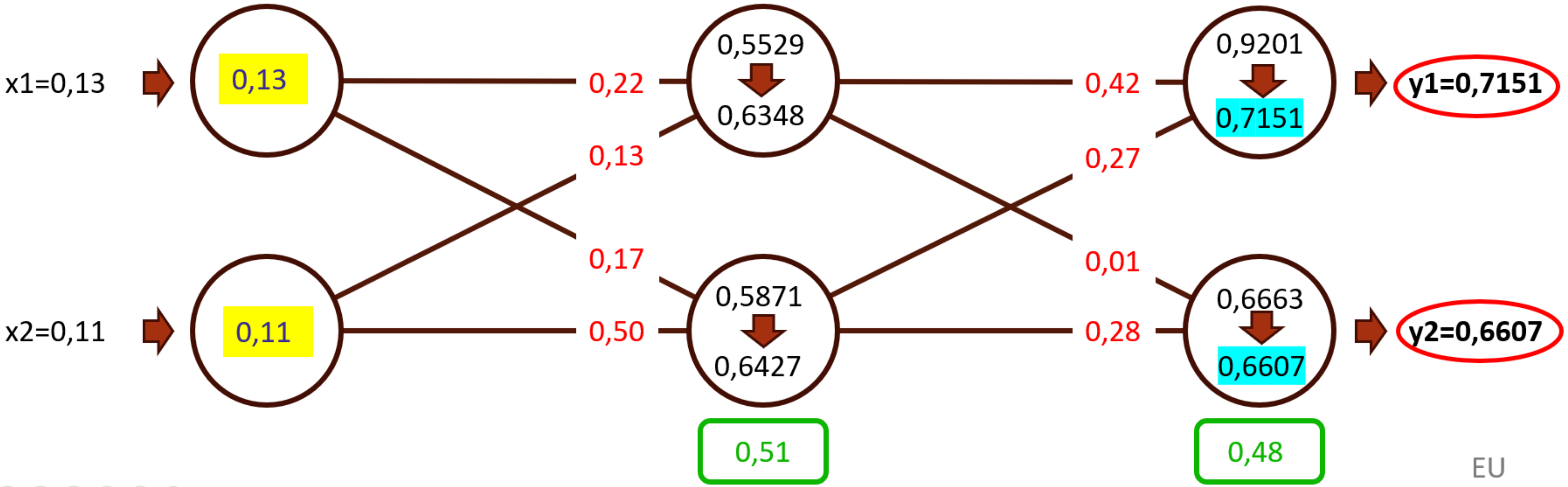
This neural network image was created by professor Timo Honkela in 1996, link (FI): http://users.ics.aalto.fi/tho/stes/step96/honkela2/



The idea of neural networks is not new, but the current neural network technology was already developed in the 1980s. The only new things are the increase in computer efficiency and the significant rise in the amount of AI training material that can be accessed on computers – it was only these advancements that made it possible to effectively utilise neural networks.

What is good to understand about the functioning of a ready-made trained neural network

- **Everything that the completed neural network has learned from the training material has been stored in the values of the numbers, i.e. calculation parameters, contained in the neural network.**
- **A neural network is just a series of simple calculations that could even be represented as an expression:**
  y1=1/(1+e^(-1*((1/(1+e^(-1*(**x1***0,22+**x2***0,13+0,51))))*0,42+(1/(1+e^(-1*(**x1***0,17+**x2***0,50+0,51))))*0,27+0,48)))
  y2=1/(1+e^(-1*((1/(1+e^(-1*(**x1***0,22+**x2***0,13+0,51))))*0,01+(1/(1+e^(-1*(**x1***0,17+**x2***0,50+0,51))))*0,28+0,48)))
- **Neural networks do not produce strictly "correct" or "incorrect" responses**; instead, they generate statistical predictions that differ in accuracy depending on their training data
- For humans, a neural network is a black box. The correctness or accuracy of the result given by the neural network cannot be checked from the network itself, but only from a source outside the network.



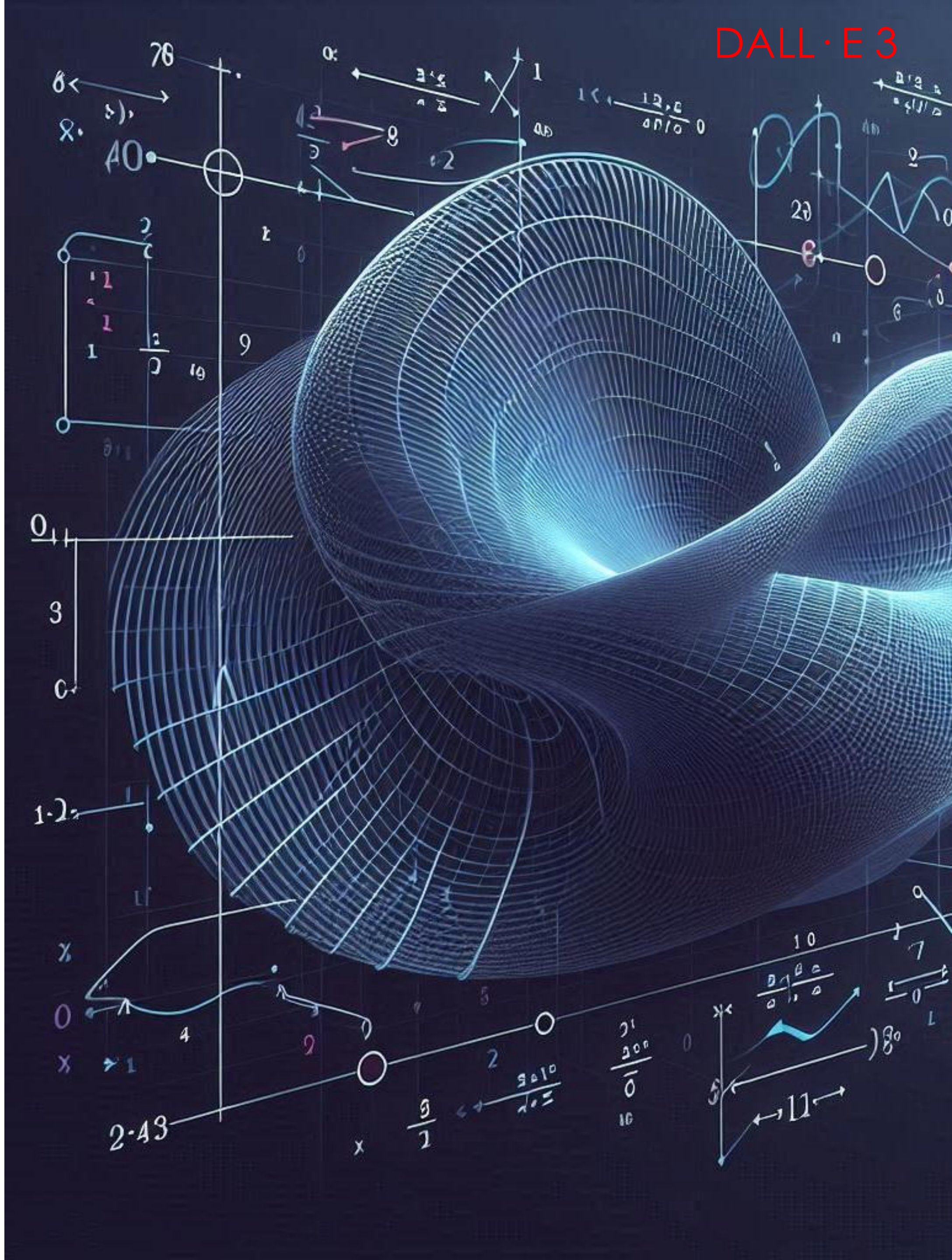
EU

Example neural network in this presentation vs real neural networks

In our example, for practical reasons, we used a small neural network with only six neurons and ten parameters. Real neural networks are typically significantly larger and can differ in many other ways from the neural network used as an example. However, what we have learned from the example neural network is also applicable to other neural networks, regardless of the network size and many other characteristics of the network.

Below are some possible differences:

- The example network is a small network of only ten parameters, but real neural networks can be as large as a billion parameters.
- Various activation functions, loss functions and other parameters that control the calculation can be used in neural networks.
- The example network is fully connected, but this is not the case for all networks.
- The example network is only connected forward, but the network can also be partially backwards-connected.



Ways of a neural network learning

- **Supervised learning,** which is based on training material and its known target outcomes, was described above. Training material can be any material from which input data and output data are known. For example,
  - a set of laboratory results and confirmed diagnoses based on them.
  - a set of images and accompanying explanations of the content of the image.
- Learning can also continue based on feedback during the use of the neural network (**reinforcement learning**). Feedback can be anything that indicates the quality or relevance of the result provided by the network, e.g.
  - Human feedback on how well an AI-generated image meets certain predetermined criteria.
  - information about the robot colliding with an obstacle.
- Learning can also take place in such a way that the algorithm independently searches for regularities in the training material and classifies it (**unsupervised learning).**