Q & A: Neural Networks

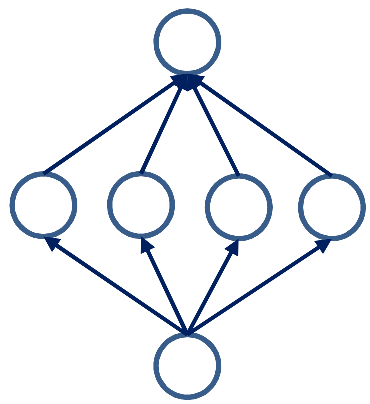
Lecture series „Machine Learning“

Niels Landwehr

Research Group „Data Science“
Institute of Computer Science
University of Hildesheim

Quiz: Number of parameters in MLP

- Assume a multilayer perceptron for regression, with a one-dimensional input and a one-dimensional output and a single hidden layer with 4 nodes:



$$f_{\boldsymbol{\theta}} : \mathbb{R} \to \mathbb{R}$$

- Question: how many model parameters does this model have, that is, what is $|\boldsymbol{\theta}|$? Remember to include the bias parameters.
  - 6 parameters
  - 7 parameters
  - 12 parameters
  - 13 parameters
  - 17 parameters
  - 20 parameters

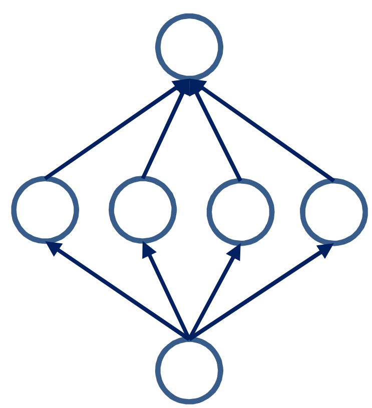
Quiz: Number of parameters in MLP

- **Solution:** the MLP has $|\boldsymbol{\theta}| = 13$ parameters
- To see that, let's look at the parameters for each layer:
  - The input layer has no parameters
  - In the intermediate layer, we have a matrix $\mathbf{W}_1 \in \mathbb{R}^{4\times 1}$ and a bias vector $\mathbf{b}_1 \in \mathbb{R}^4$, so eight parameters overall
  - In the final layer, we have a matrix $\mathbf{W}_2 \in \mathbb{R}^{1\times 4}$ and a bias term $\mathbf{b}_2 \in \mathbb{R}^1$, so five parameters overall

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}_2 \mathbf{z}_1 + \mathbf{b}_2 \qquad \mathbf{W}_2 \in \mathbb{R}^{1\times 4}, \quad \mathbf{b}_2 \in \mathbb{R}^1$$

$$\mathbf{z}_1 = \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \qquad \mathbf{W}_1 \in \mathbb{R}^{4\times 1}, \quad \mathbf{b}_1 \in \mathbb{R}^4$$

$$\mathbf{x} \in \mathbb{R}^M$$

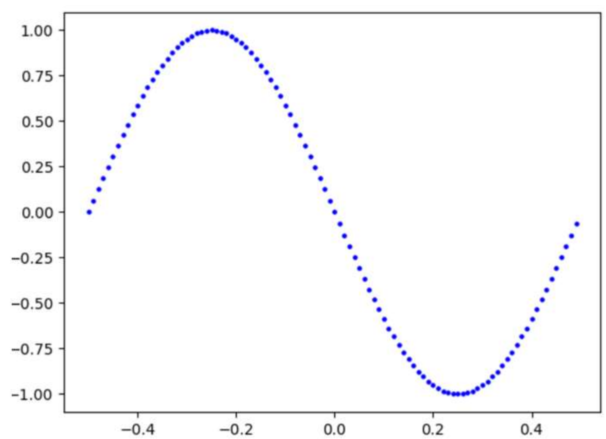


Quiz: Fitting the MLP to toy data

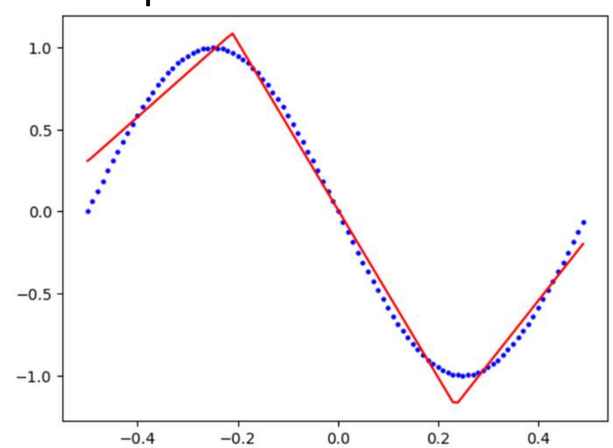
- We can use the MLP from the previous slide to solve a simple toy regression problem
  - Inputs: 100 data points uniformly spaced in interval [-0.5,0.5]
  - Targets: obtained from sine function, $y = \sin((x+0.5)\cdot 2\pi)$

- Fitting the MLP from last slide using squared loss to this data set results in:

data



data and prediction function of model



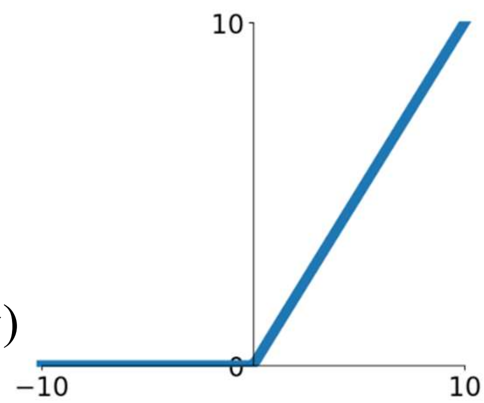
- **Question**: which nonlinear activation function did I use? ReLU, sigmoid, or tanh?

Quiz: Fitting the MLP to toy data

- **Solution**: ReLU activations.
  - For ReLU activations, the only operations used in the model are linear operations and maximum
  - Therefore, the final function stays piecewise linear

ReLU: Rectified linear unit

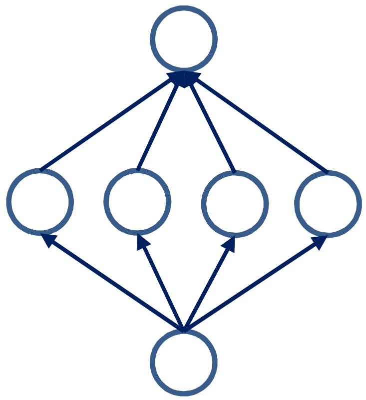
$$\sigma(z) = \max(0, z)$$



$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}_2 \mathbf{z}_1 + \mathbf{b}_2$$

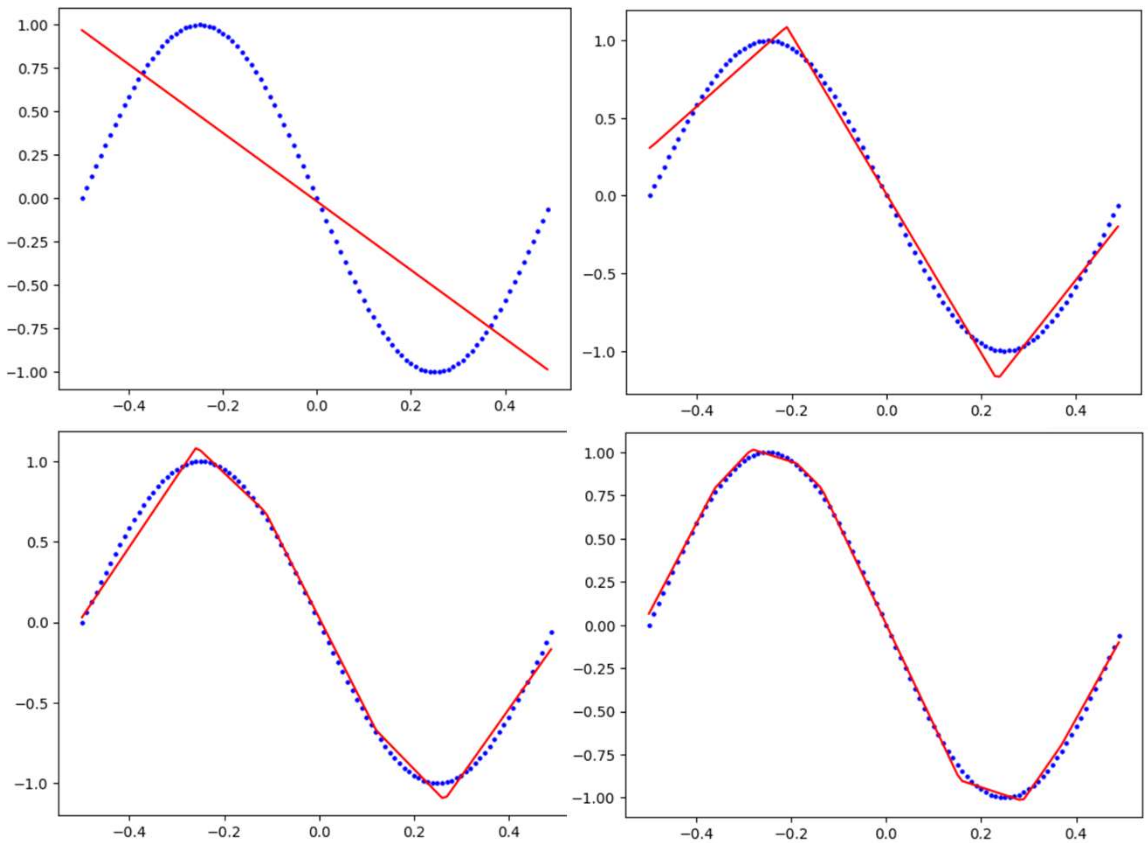
$$\mathbf{z}_1 = \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)$$

$$\mathbf{x} \in \mathbb{R}^M$$



Quiz: Fitting MLP to toy data

- We now study variants of the MLP. In the figures below, we see results of fitting four different variants of the MLP given above to the same sine data set:



Question: what was changed between the different variants?

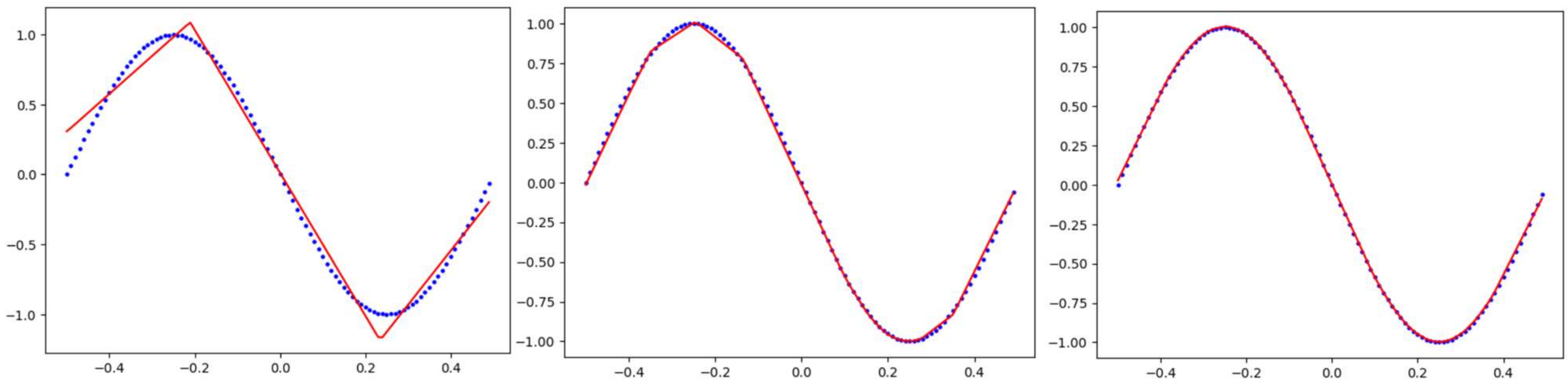
– The number of nodes in the hidden layer: 2, 4, 40, 400
– The nonlinear activation function: leaky ReLU, ReLu, sigmoid, tanh
– The number of hidden layers: 0, 1, 2, 3 hidden layers
– The regularizer of the model

Quiz: Fitting the MLP to toy data

- **Solution**: Number of hidden layers changes
  - This is most easily seen in the first plot: without a hidden layer, the model is linear
  - The more hidden layers we have, the more expressive/flexible the model becomes, so it can fit the training data more accurately

Quiz: Fitting MLP to toy data

- We now study variants of the MLP. In the figures below, we see results of fitting four different variants of the MLP given above to the same sine data set:



Question: what was changed between the different variants?

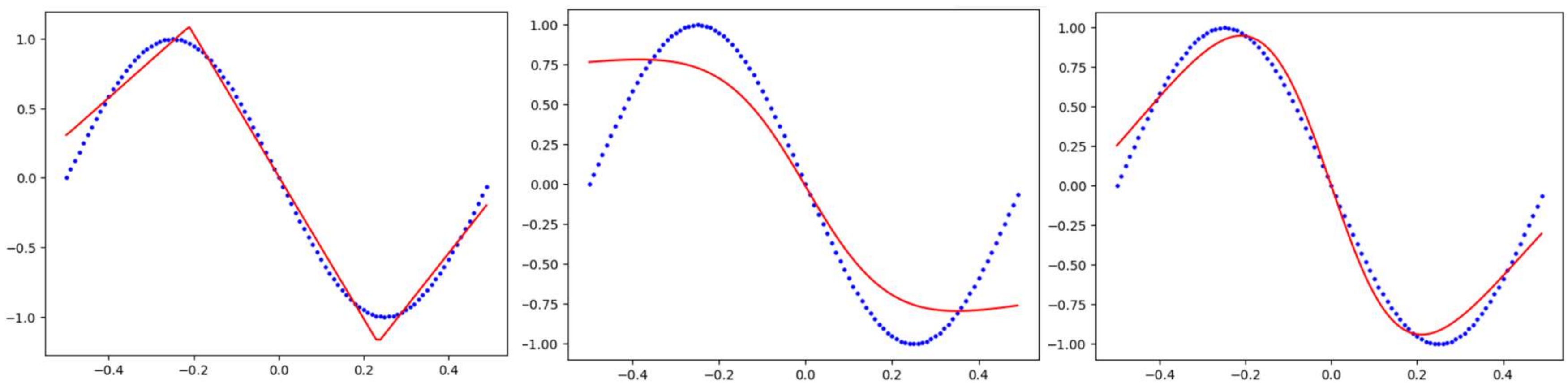
- The number of nodes in the hidden layer: 4, 40, 400
- The nonlinear activation function: ReLU, sigmoid, tanh
- The regularizer of the model
- The number of data points

Quiz: Fitting the MLP to toy data

- **Solution**: Number of nodes in hidden layer changes (4, 40, 400 nodes)
  - As the number of nodes changes, the model becomes more expressive/flexible and therefore can fit the training data more accurately
  - Changing the activation function or the regularizer or the number of data points would not affect the representational capacity of the model
  - Note that the model stays piecewise linear, but with more segments in the piecewise linear function, making it appear smoother

Quiz: Fitting MLP to toy data

- We now study variants of the MLP. In the figures below, we see results of fitting four different variants of the MLP given above to the same sine data set:



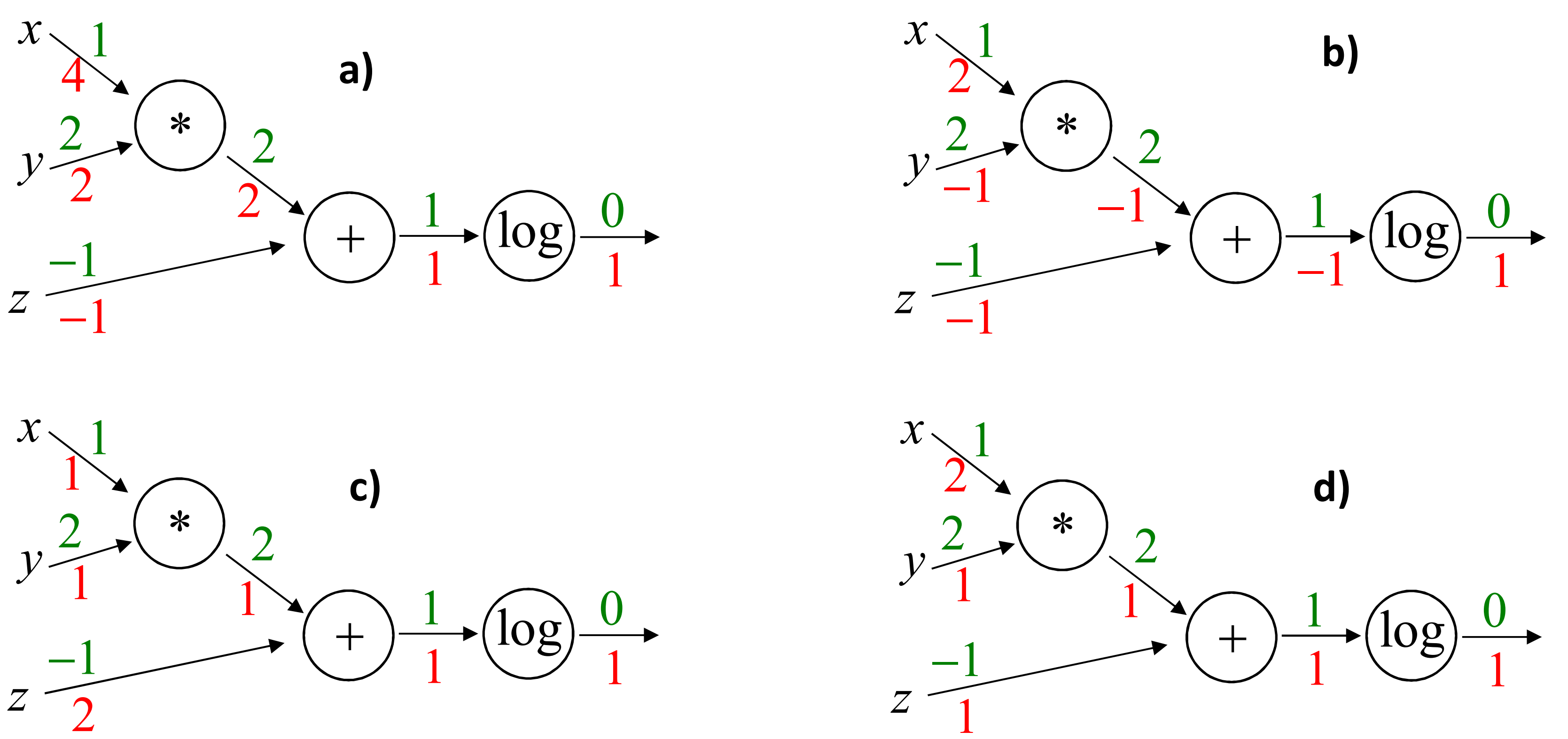
Question: what was changed between the different variants?
– The number of nodes in the hidden layer: 4, 40, 400
– The nonlinear activation function: ReLU, sigmoid, tanh
– The regularizer of the model
– The number of data points

Quiz: Fitting the MLP to toy data

- **Solution**: The activation function (ReLU, sigmoid, tanh)
  - In the second and third plot, the functions are smooth (not piecewise linear anymore)
  - However, the number of parameters and therefore the representational capacity of the model has not really increased: the learned function is smoother but not really a better fit to training data

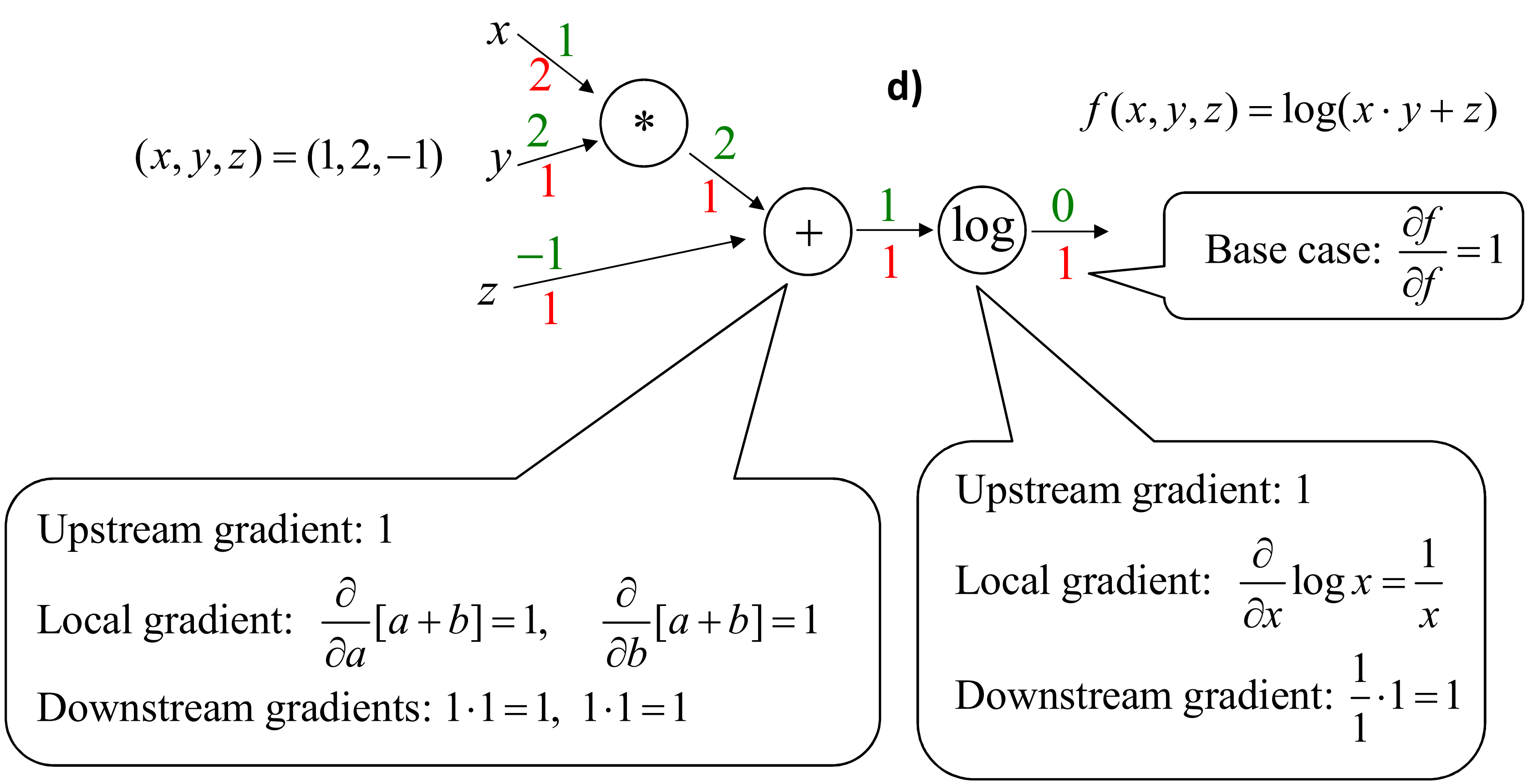
Quiz: Automatic Differentiation

- Assume $f:\mathbb{R}^3 \rightarrow \mathbb{R}$ with $f(x,y,z) = \log(x \cdot y + z)$, where log denotes natural logarithm
- **Question**: Which is the correct compute graph and corresponding backpropagation result for the gradient ∇f at the point $(x,y,z) = (1,2,-1)$?



Quiz: Automatic Differentiation

- **Solution**: the correct backpropagation procedure is d)



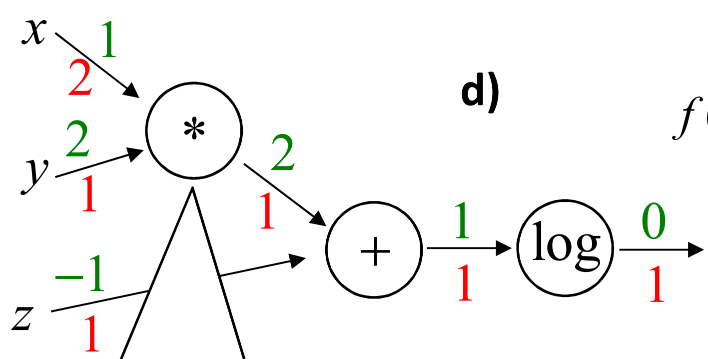
Quiz: Automatic Differentiation

- **Solution**: the correct backpropagation procedure is d)

$(x, y, z) = (1, 2, -1)$

d)

$f(x, y, z) = \log(x \cdot y + z)$



Upstream gradient: 1

Local gradient: $\frac{\partial}{\partial a}[a \cdot b] = b, \quad \frac{\partial}{\partial b}[a \cdot b] = a$

Downstream gradients: $1 \cdot 2 = 1,\ \ 1 \cdot 1 = 1$