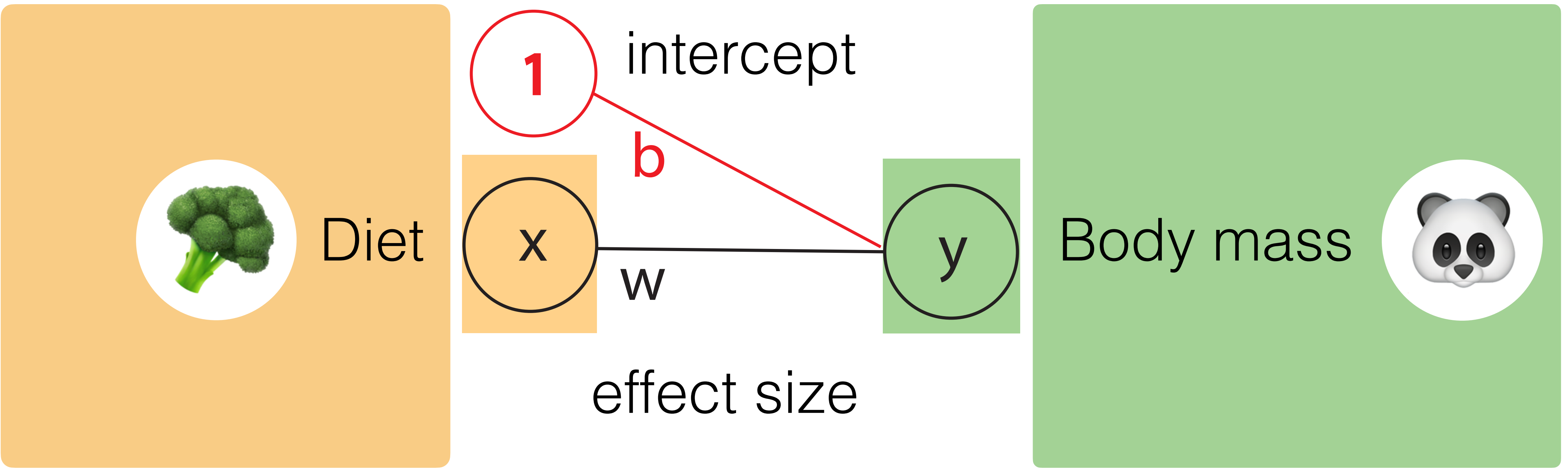
Neural networks: (very) short version…

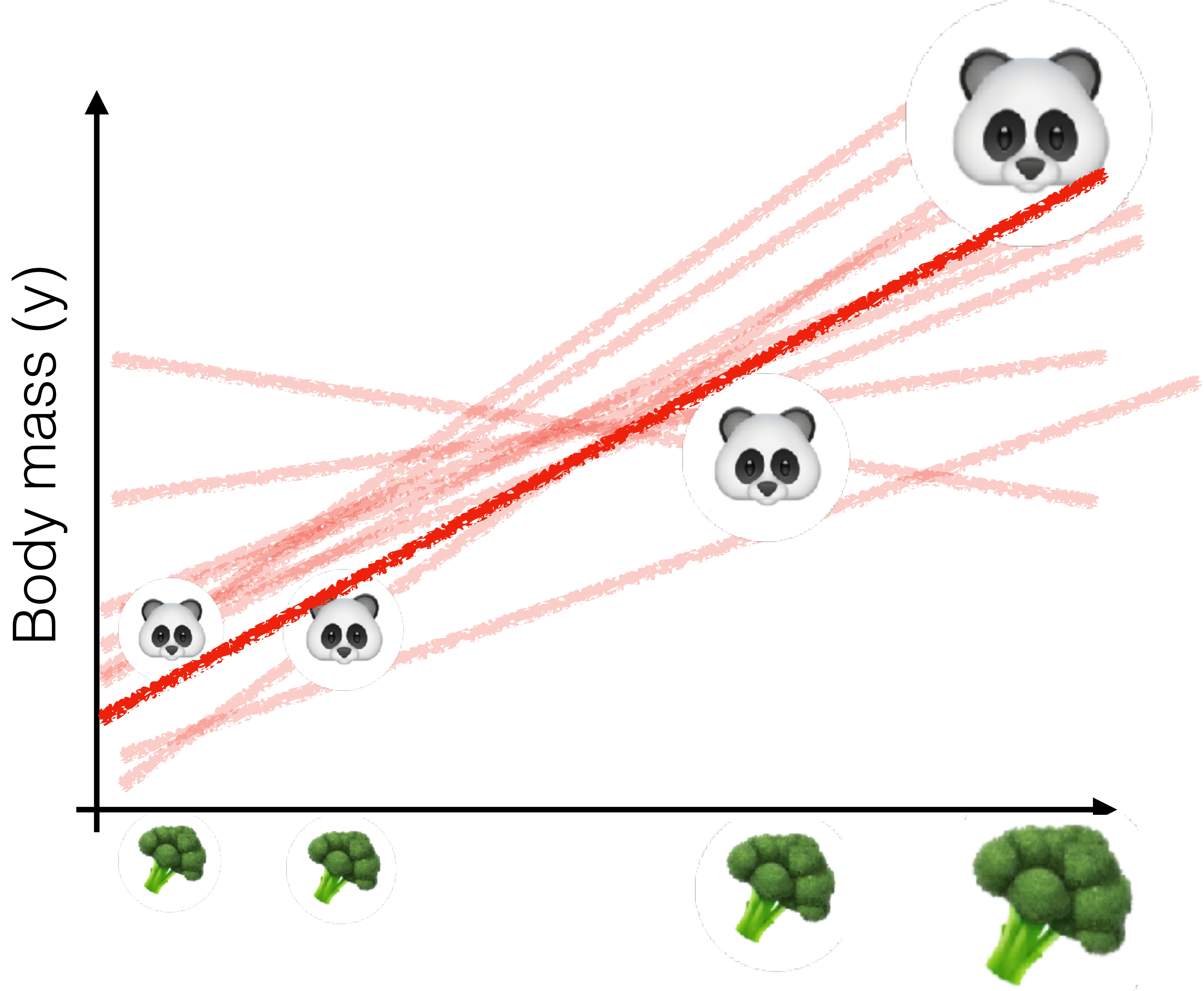
Linear regression



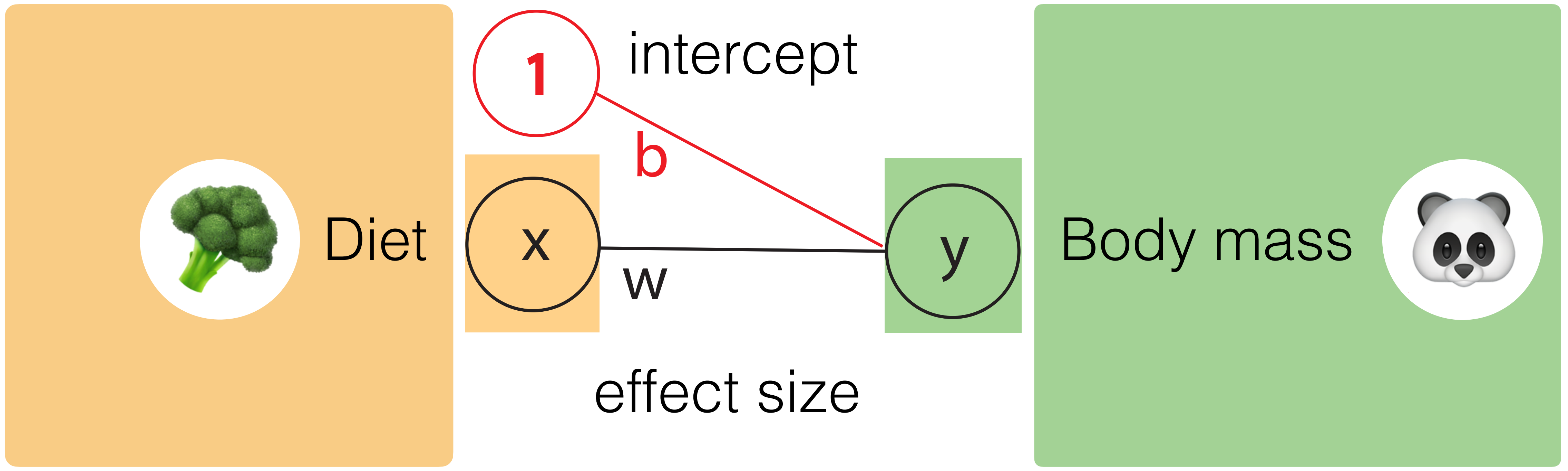
$$y = wx + b$$

prediction effect size intercept

Optimizing ('training') a model



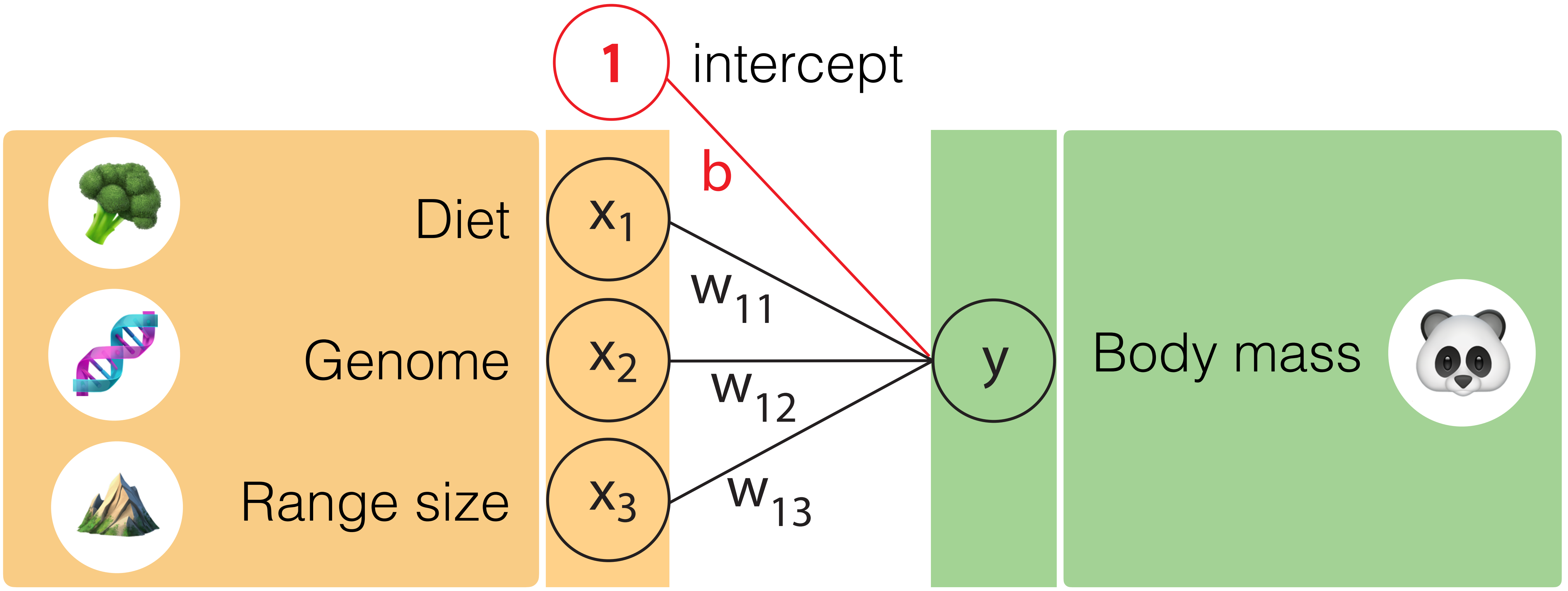
Linear regression



$$y = wx + b$$

prediction   effect size   intercept

Multiple linear regression



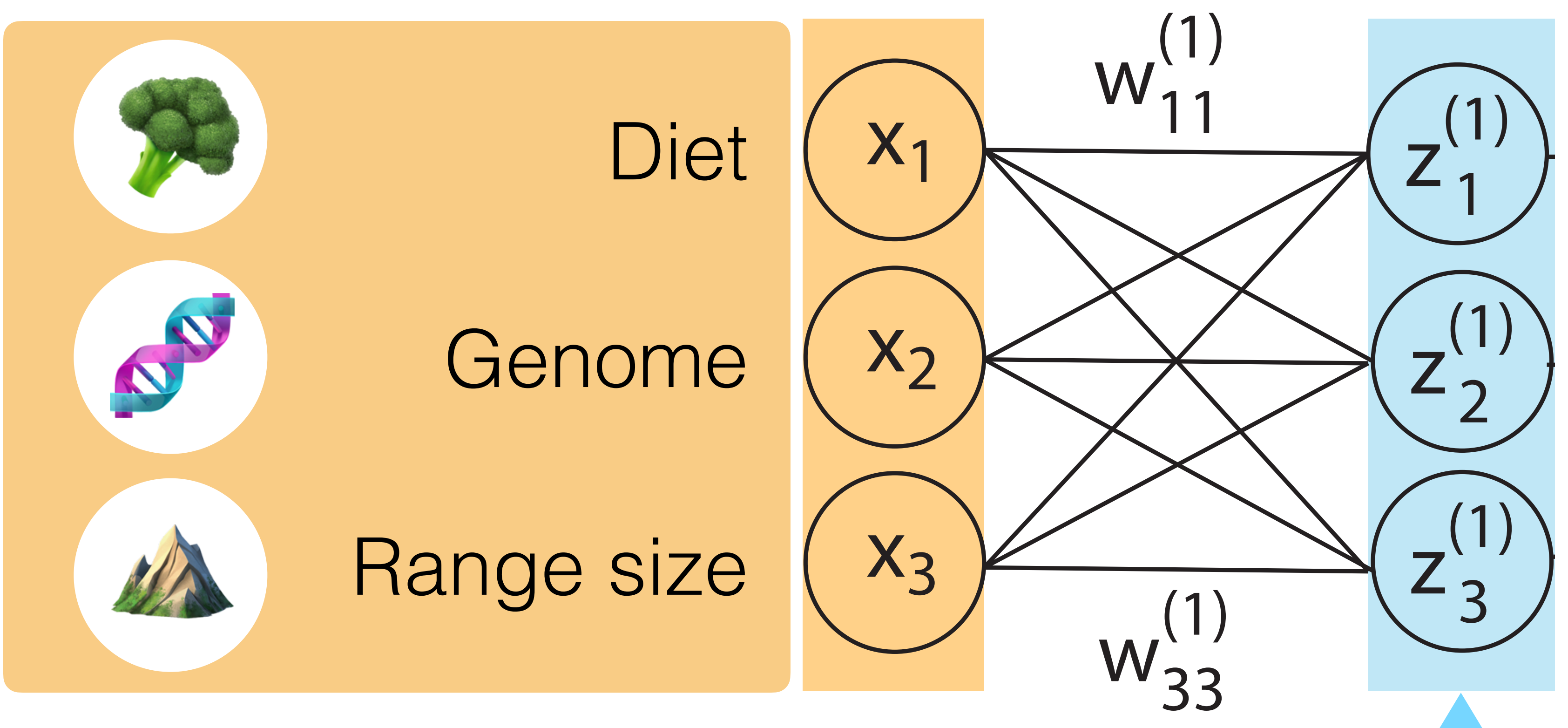
$$y = w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + b$$

prediction   effect sizes   intercept

$$y = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \begin{pmatrix} w_{11} & w_{12} & w_{13} \end{pmatrix} + b$$

We can re-write it as a matrix multiplication

Neural network



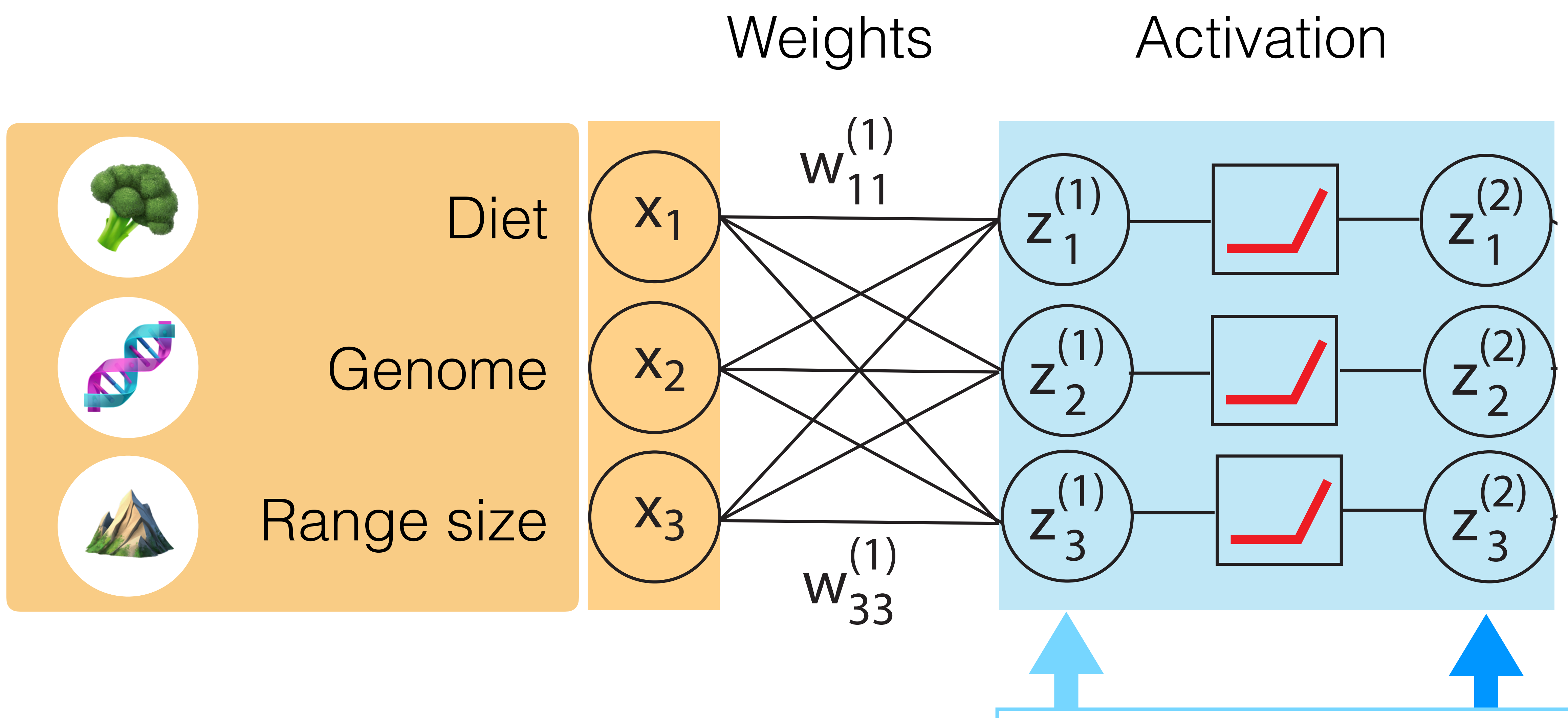
$$z_i^{(1)} = w_{i1}^{(1)} x_1 + w_{i2}^{(1)} x_2 + w_{i3}^{(1)} x_3$$

Intermediate (hidden) values $z^{(1)}$: linear function of all predictors

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} = \begin{pmatrix} z_1^{(1)} \\ z_2^{(1)} \\ z_3^{(1)} \end{pmatrix}$$

x $\quad$ $w^{(1)}$ $\quad$ $z^{(1)}$

Neural network



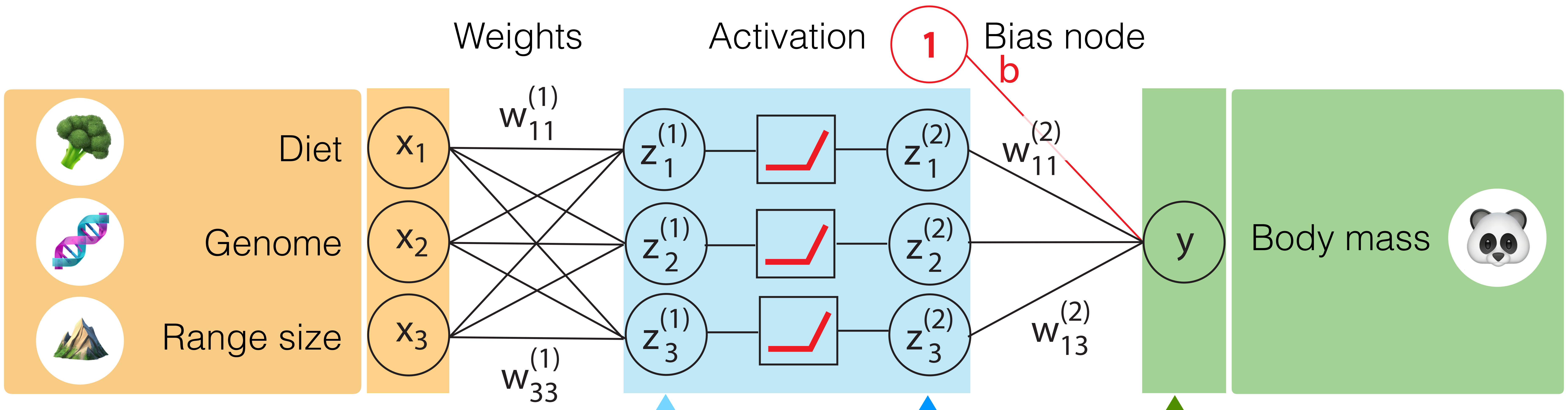
$$z_i^{(1)} = w_{i1}^{(1)} x_1 + w_{i2}^{(1)} x_2 + w_{i3}^{(1)} x_3$$

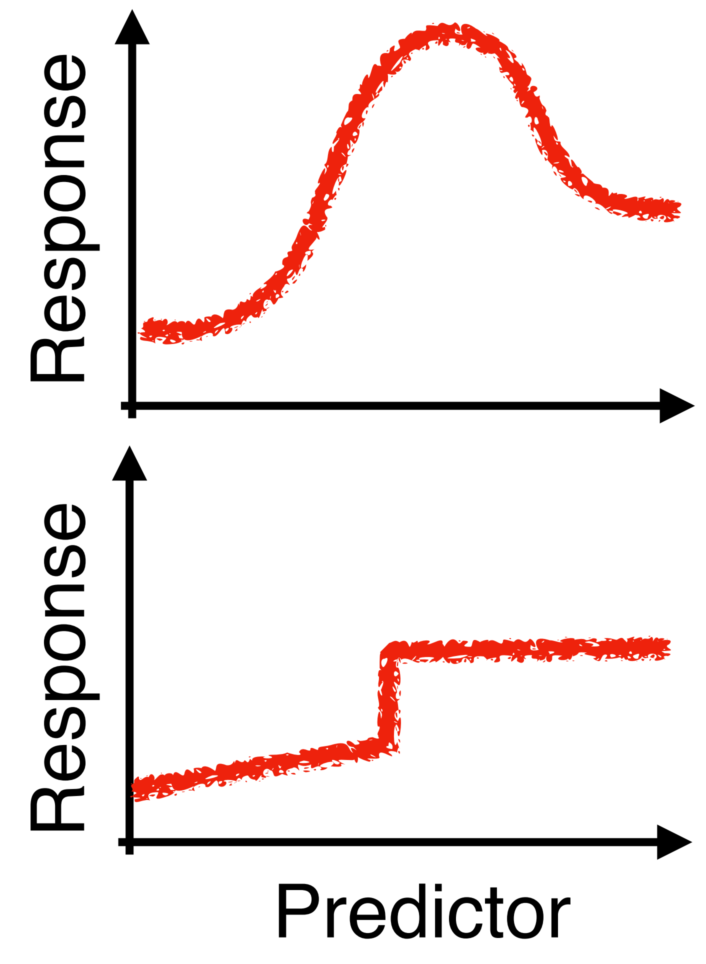
Intermediate (hidden) values $z^{(1)}$: linear function of all predictors

$$z_i^{(2)} = \max(0, z_i^{(1)})$$

Intermediate (hidden) values $z^{(2)}$: non-linear function of $z^{(1)}$

Neural network





$$z_i^{(1)} = w_{i1}^{(1)} x_1 + w_{i2}^{(1)} x_2 + w_{i3}^{(1)} x_3$$

Intermediate (hidden) values $z^{(1)}$: linear function of all predictors

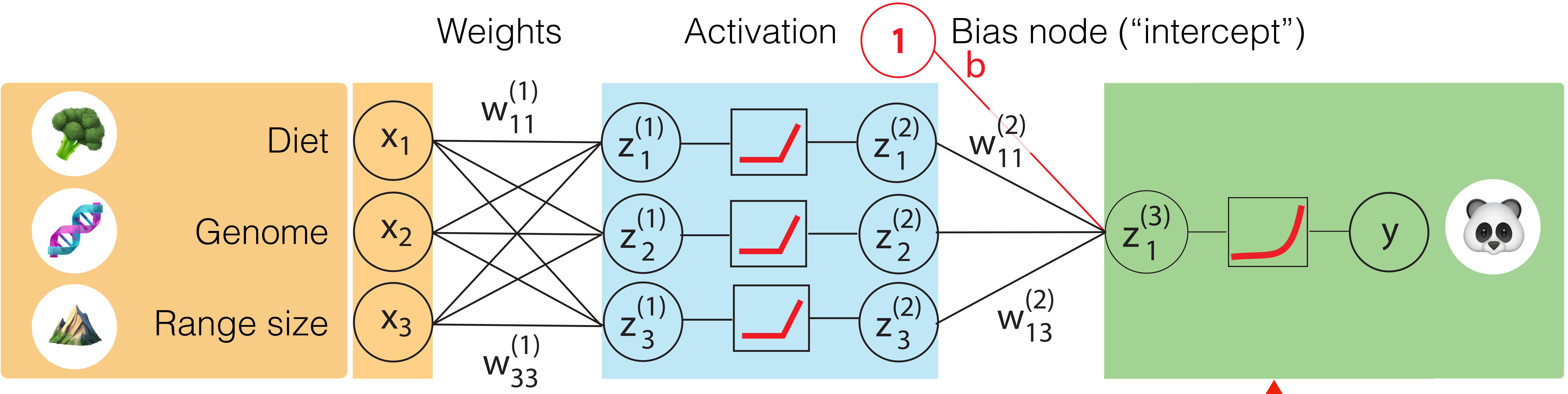
$$z_i^{(2)} = \max(0, z_i^{(1)})$$

Intermediate (hidden) values $z^{(2)}$: non-linear function of $z^{(1)}$

$$y = w_{11}^{(2)} z_1^{(2)} + w_{12}^{(2)} z_2^{(2)} + w_{13}^{(2)} z_3^{(2)} + b$$

Output: linear function of $z^{(2)}$

Neural network



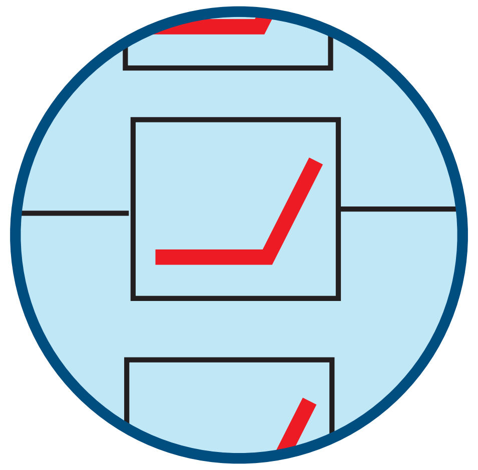
$y = e^z$ ← Must be positive

Add an output activation function to match expected range

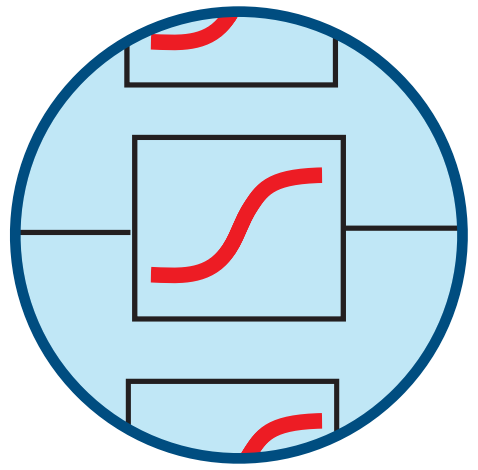
$y = 1/(1 + e^{-z})$ ← Must be in [0, 1]

Parameterization of a neural network

Activation functions



ReLU: rectified linear unit

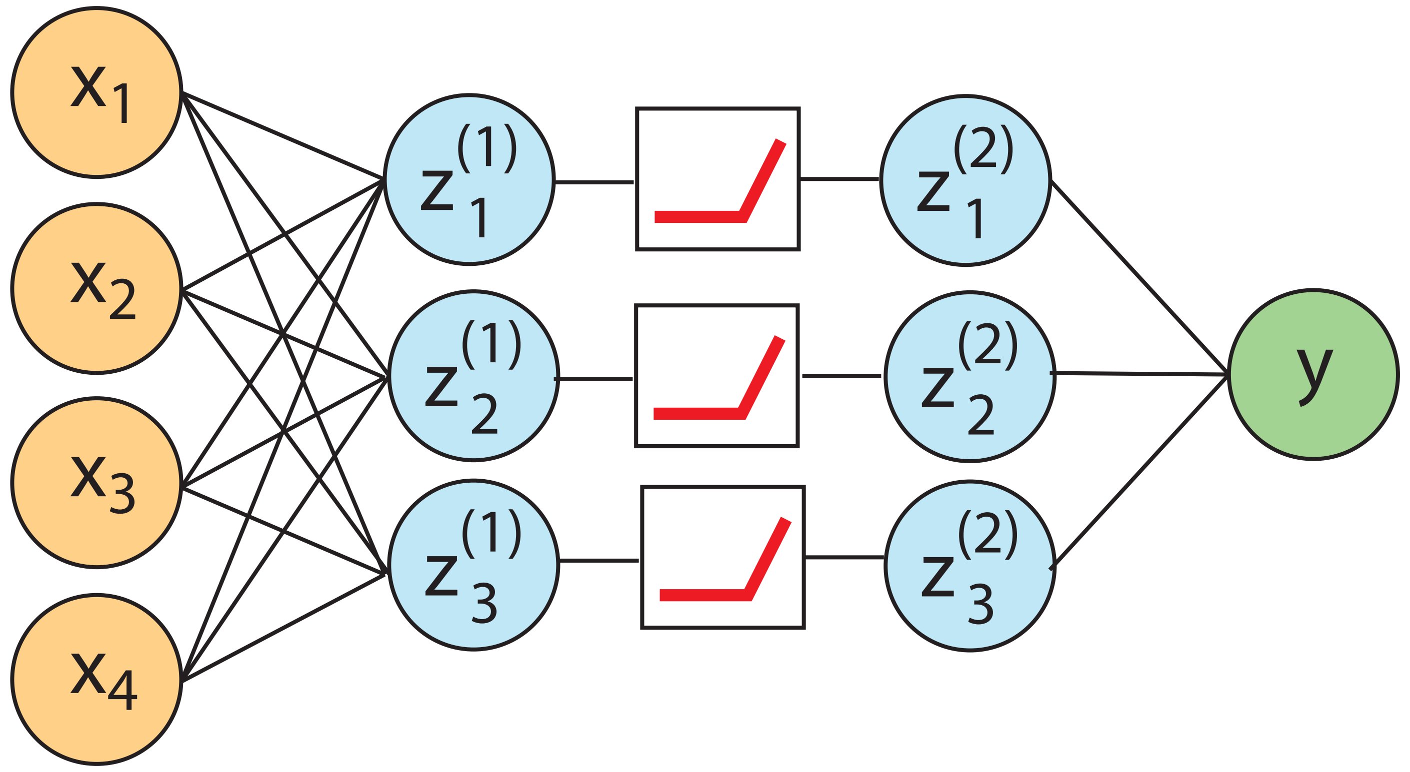


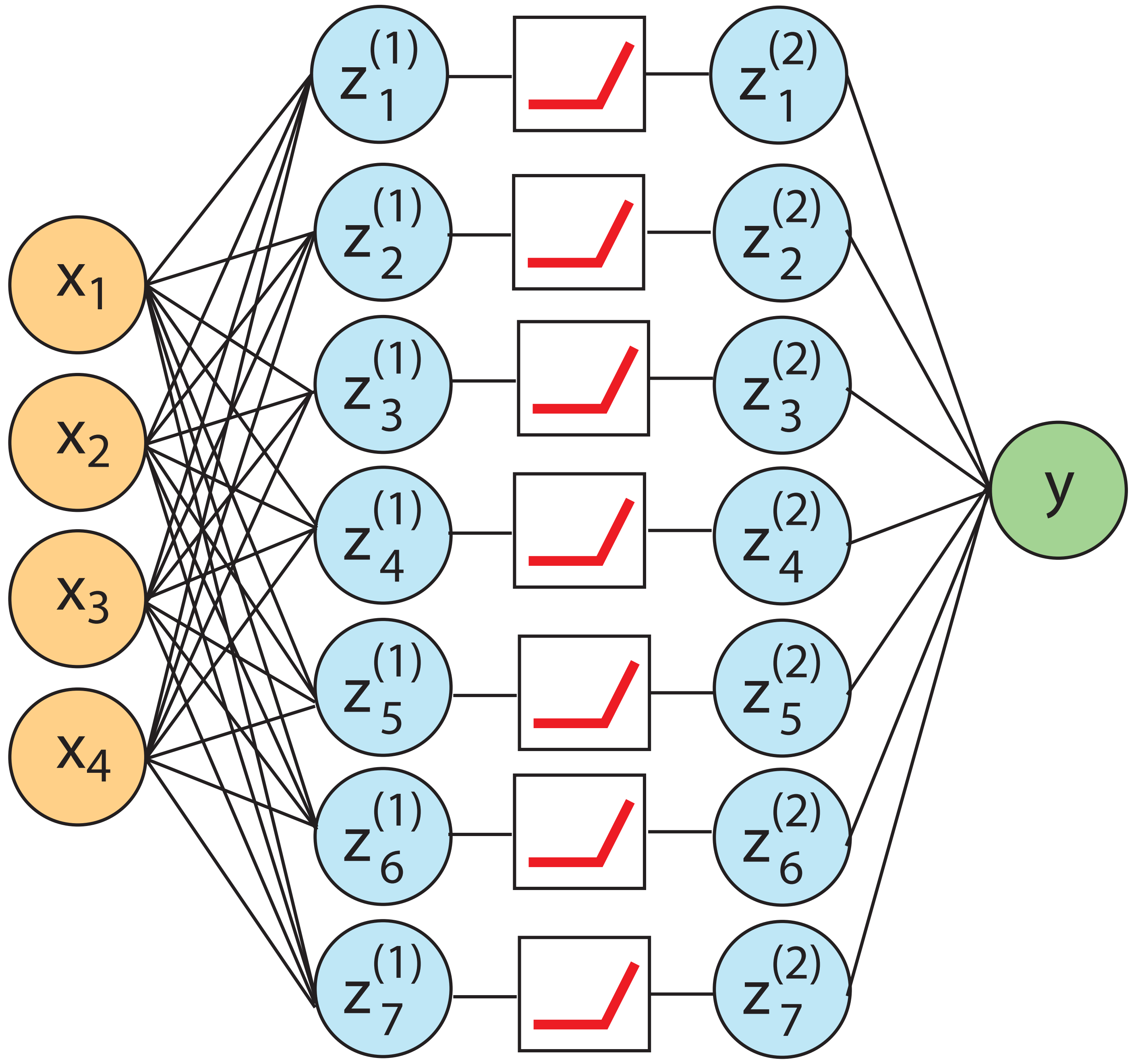
Sigmoid

```
ReLU(x) = max(0, x)
```

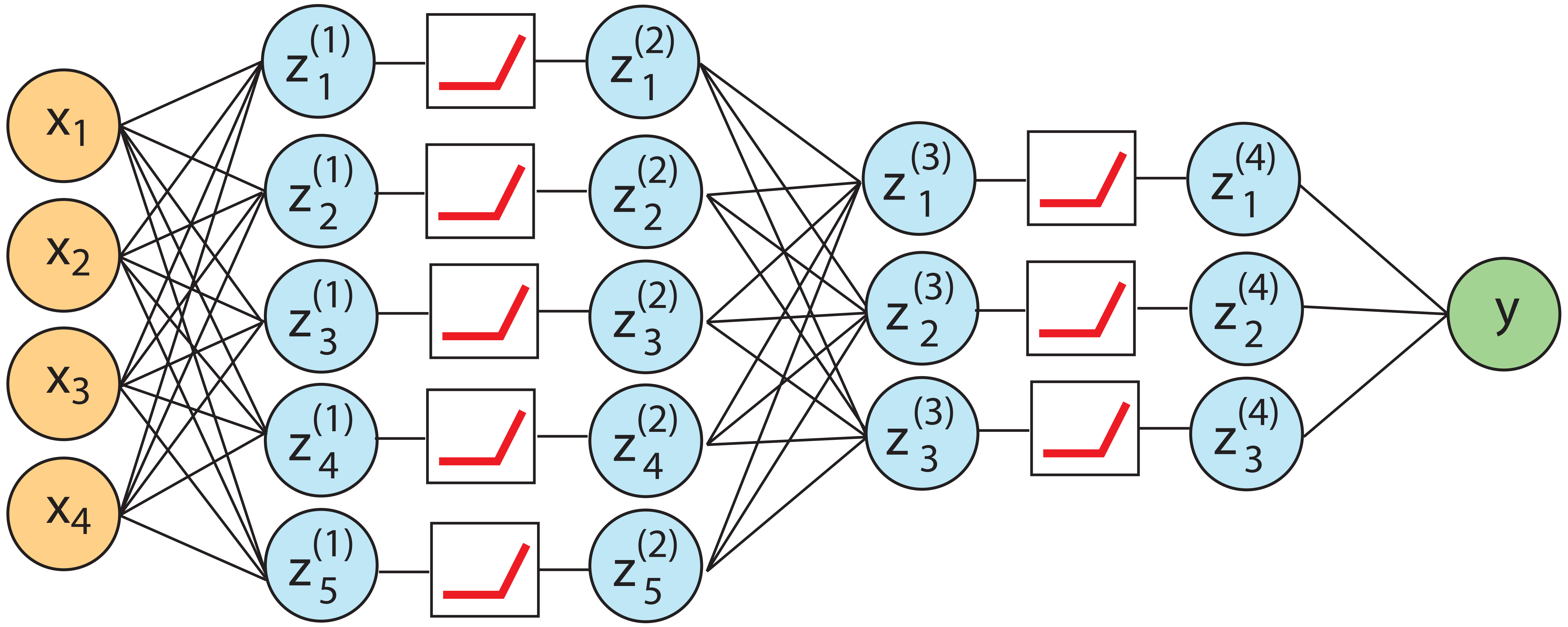
```
sigmoid(x) = 1 / (1 + exp(-x))
```

Number of nodes

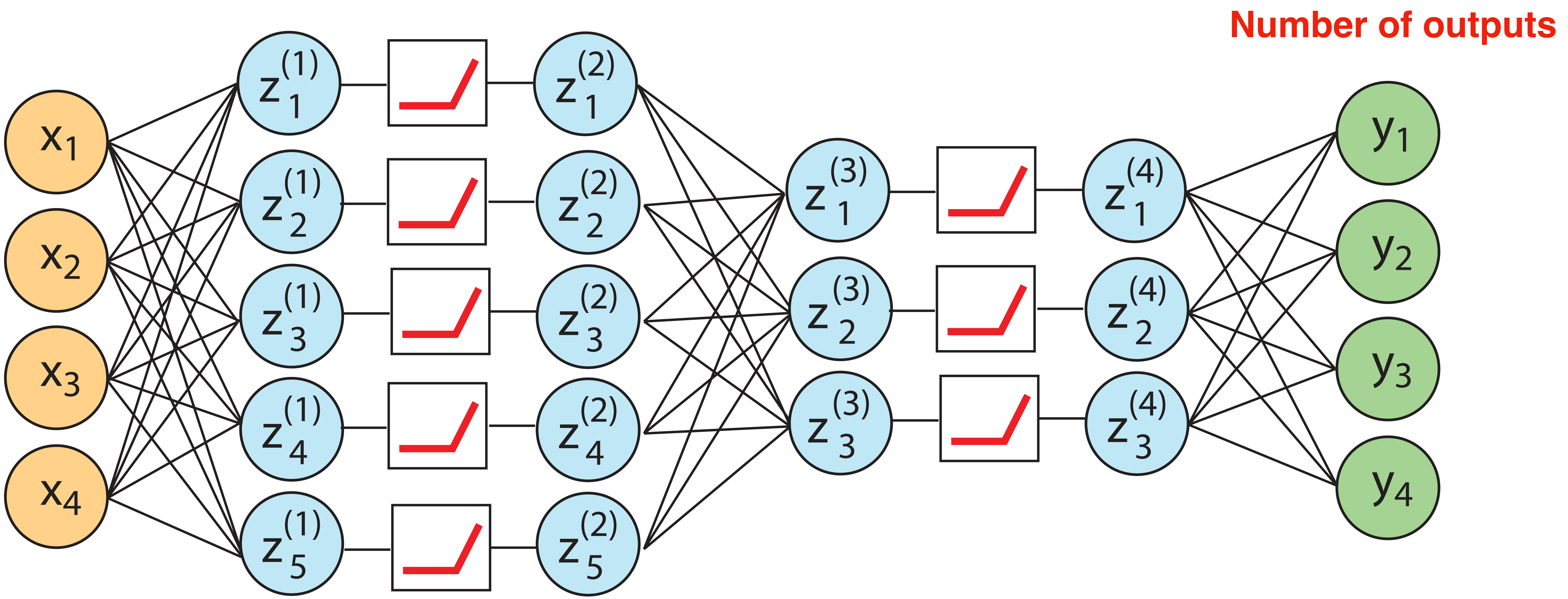




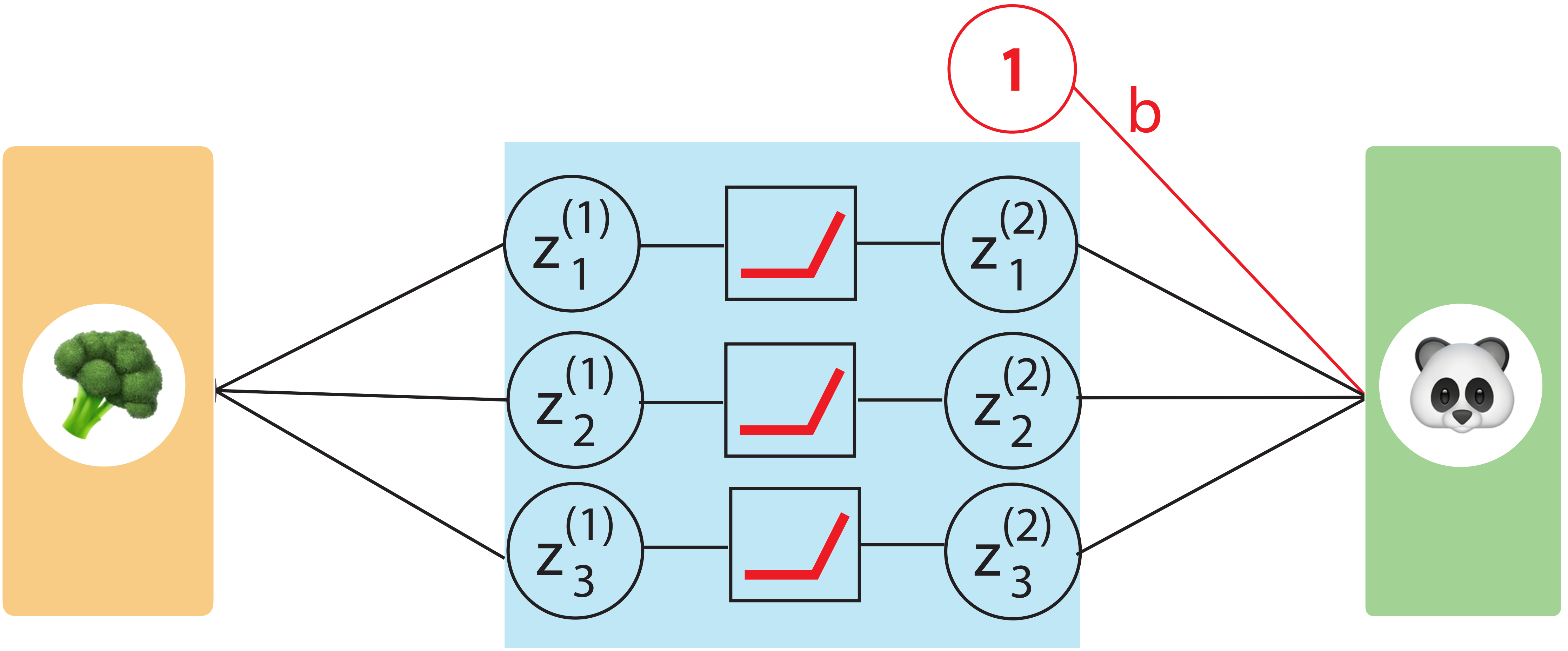
Number of hidden layers (deep NNs)



Parameterization of a neural network

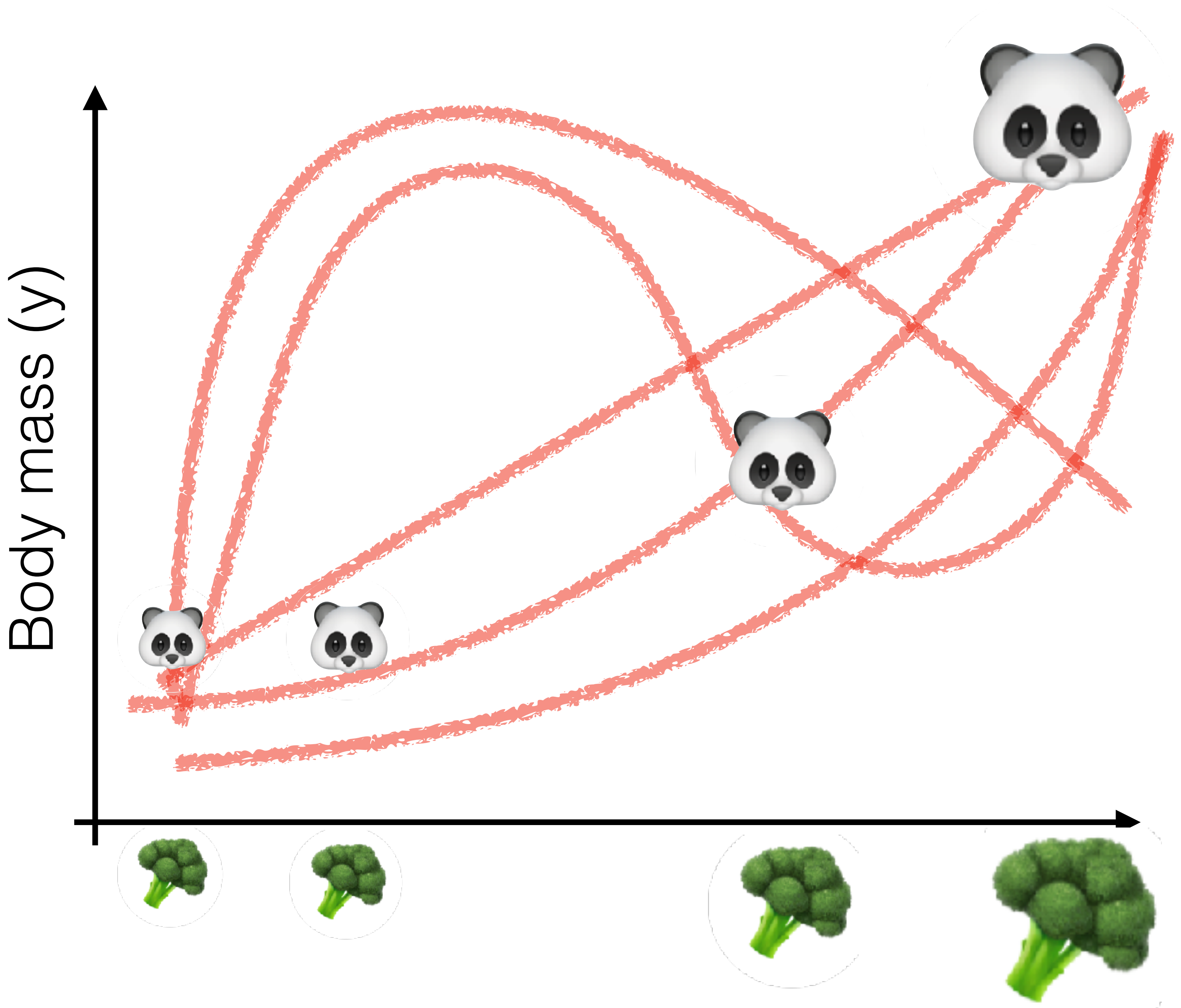


NN regression



Data ('training set')

Optimizing ('training') a model



Likelihood function: the normal density function

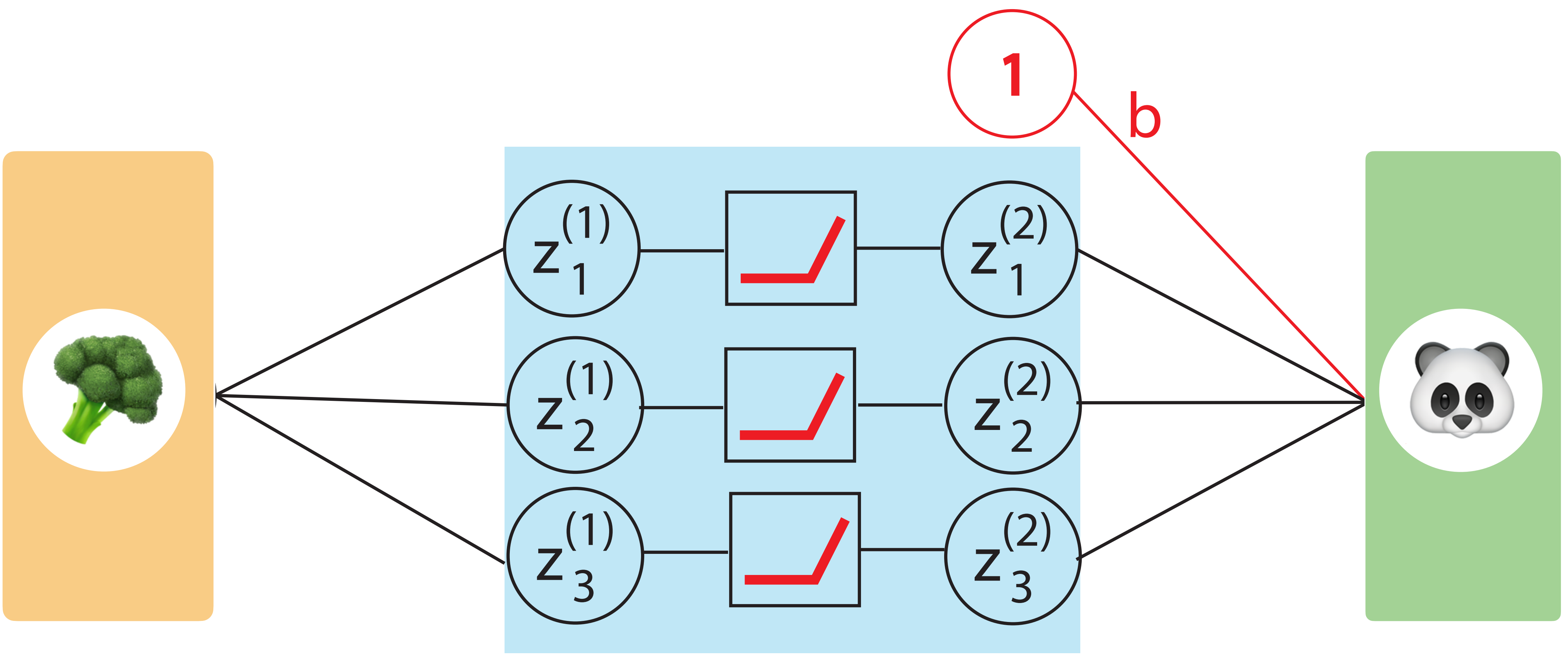
$$P(\text{🐼} \mid \text{y}, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\text{🐼} - \text{y})^2}{2\sigma^2}}$$

Loss function: mean squared error

$$-\log P(\text{🐼} \mid \text{y}) \propto (\text{🐼} - \text{y})^2$$

Squared difference between truth and prediction

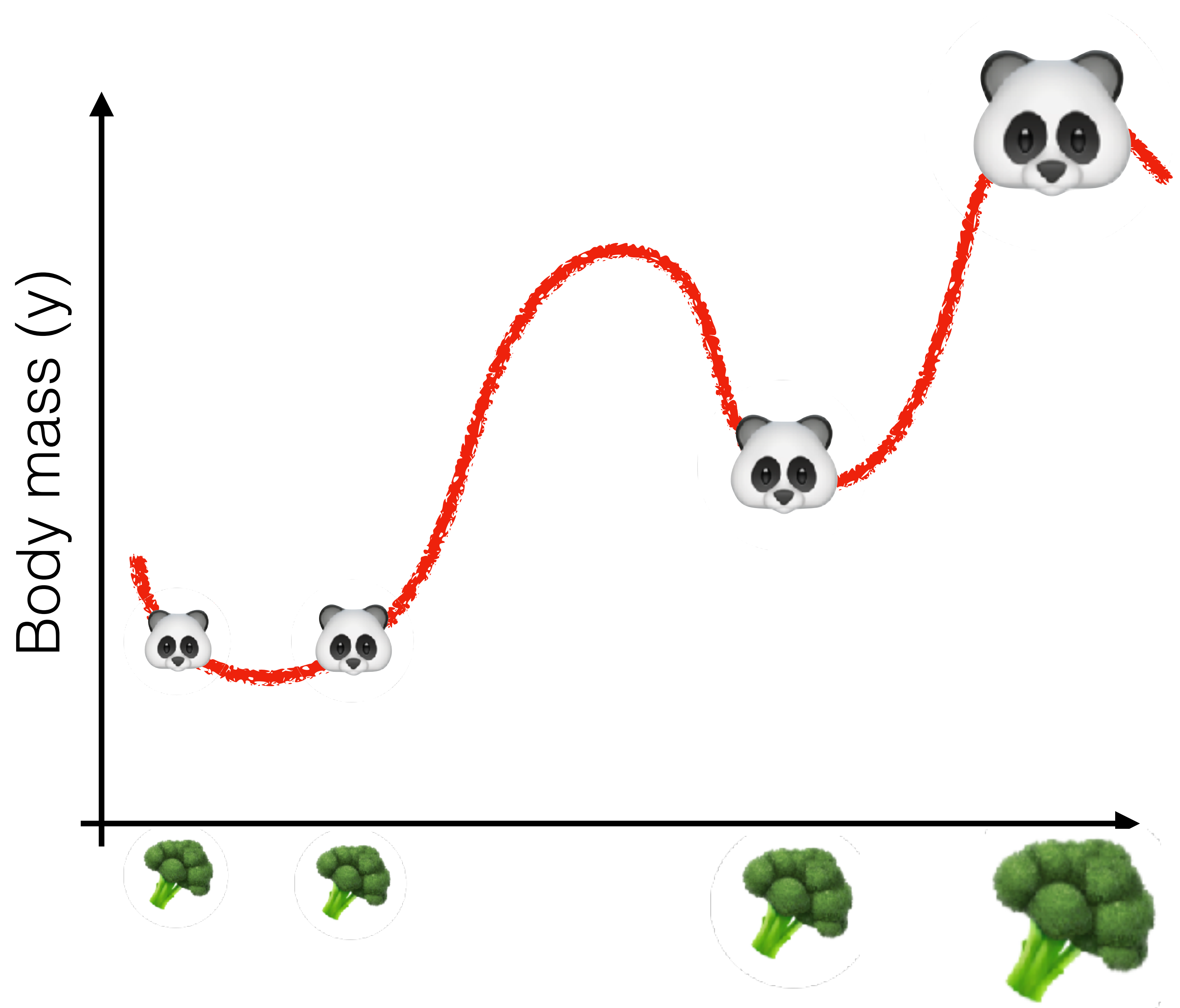
NN regression



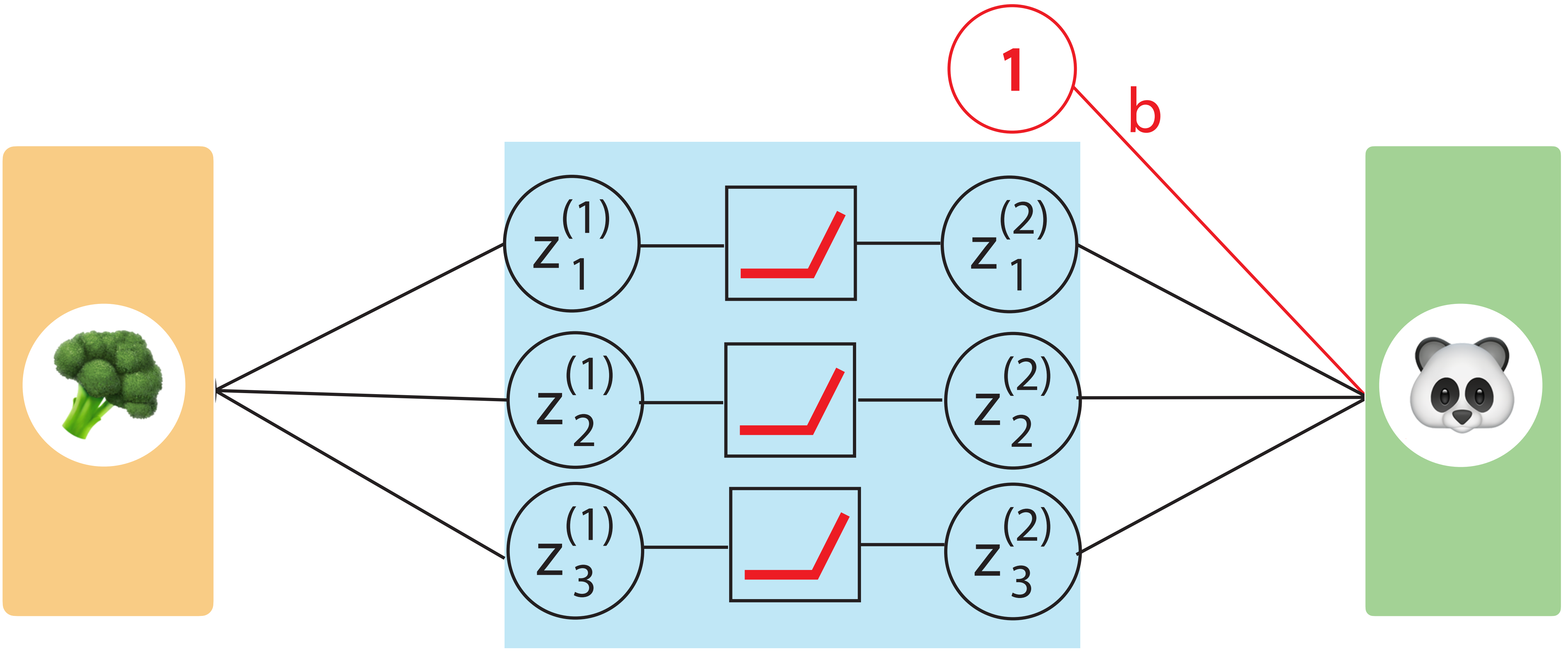
Neural networks are over-parameterized models
and will overfit with a maximum likelihood approach

Data ('training set')

Best-fitting model

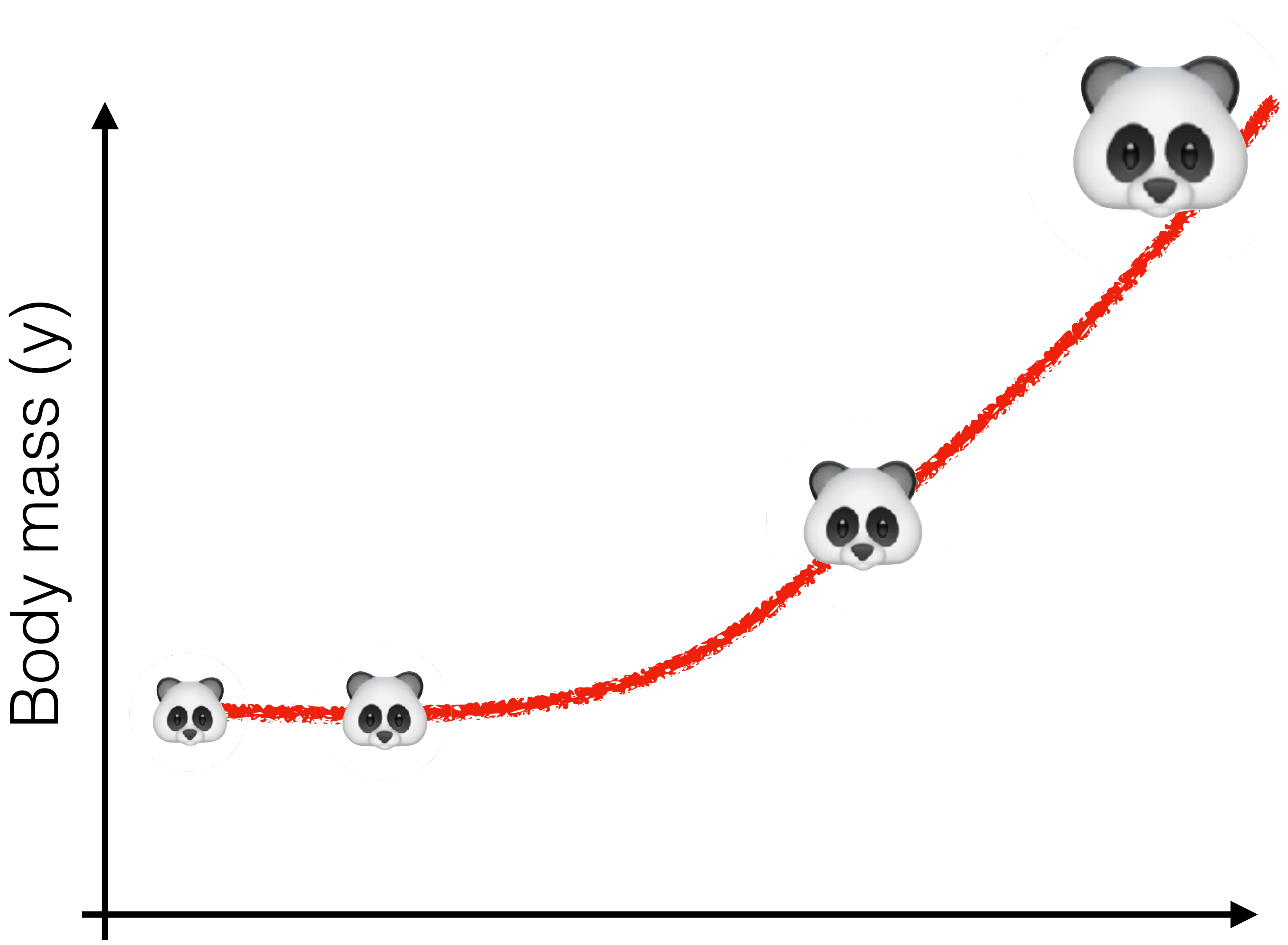


NN regression



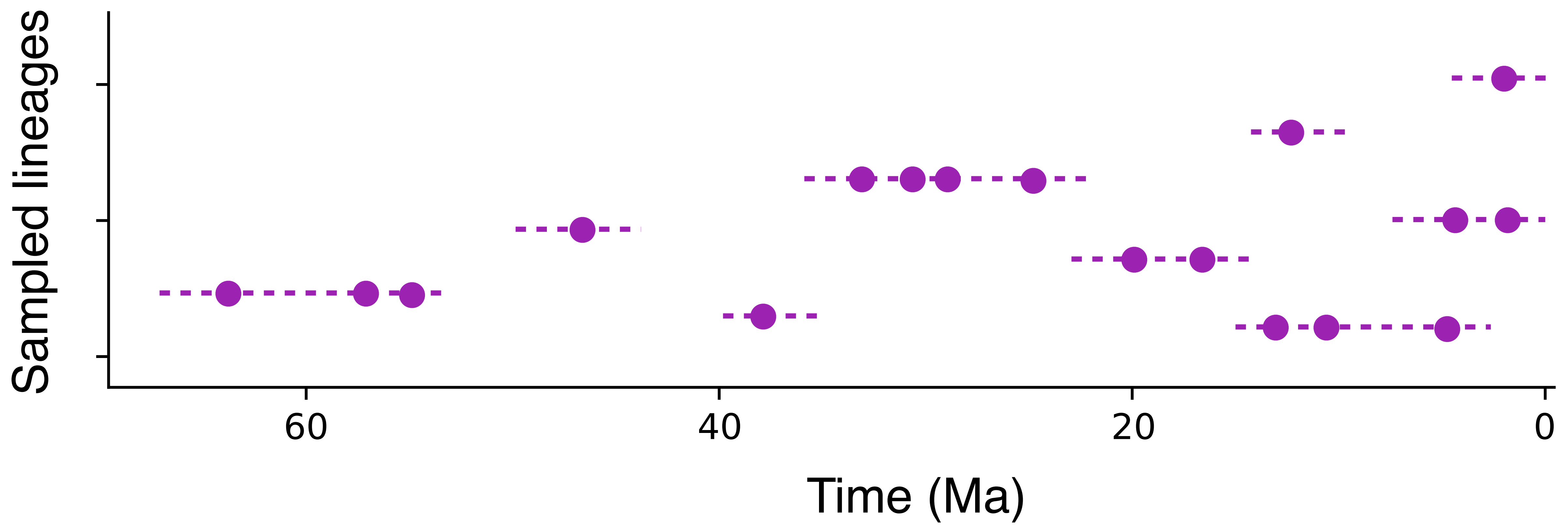
Data ('training set')

Trained model

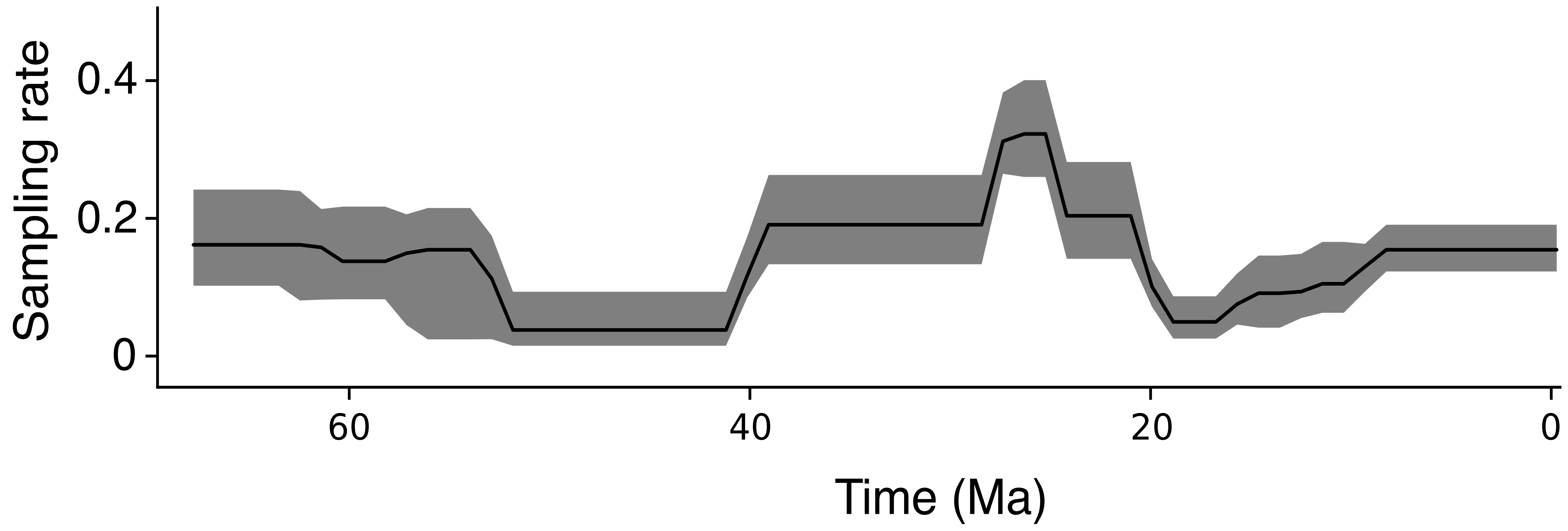


Bayesian (unsupervised) estimation of speciation and extinction rates from fossils

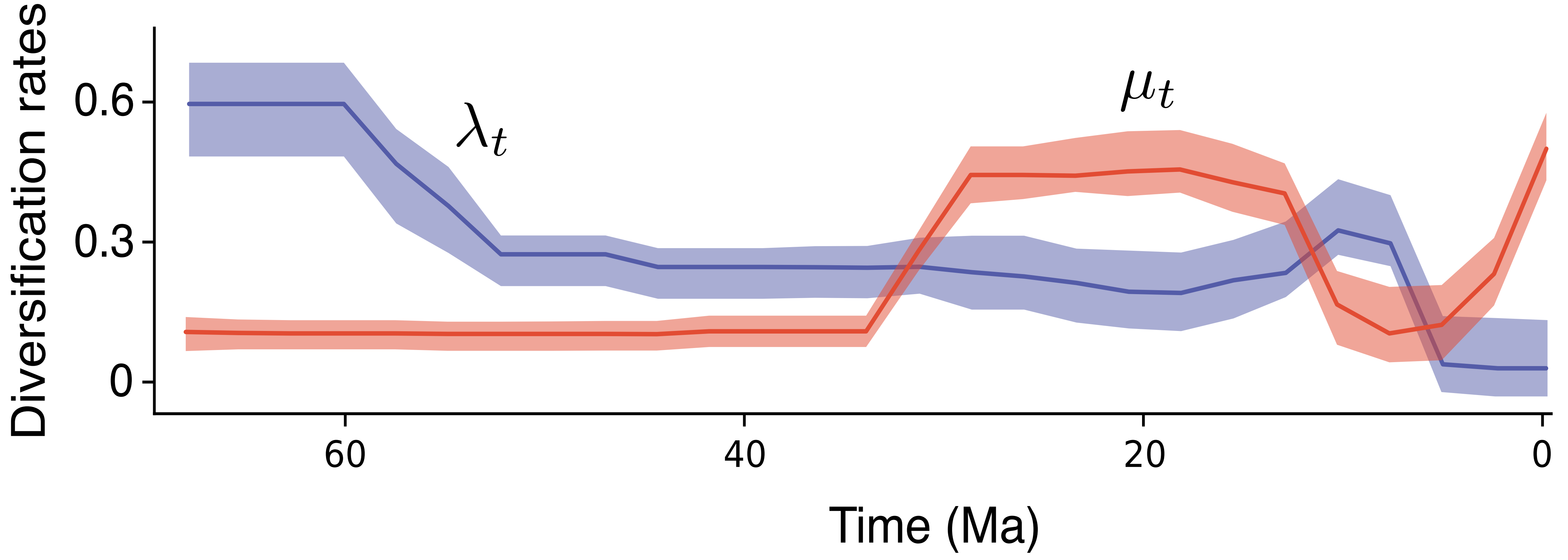
Fossil record



Preservation rates
(Poisson process)



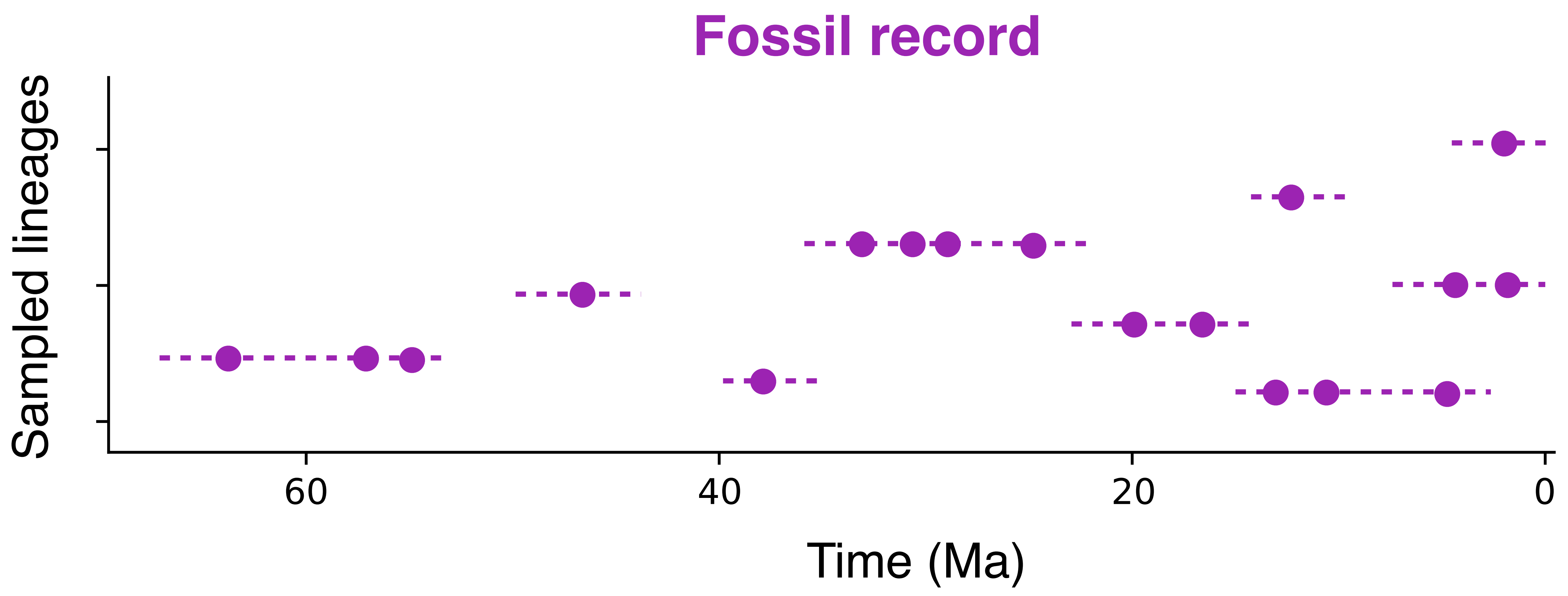
Speciation & **extinction** rates
(Birth-death process)



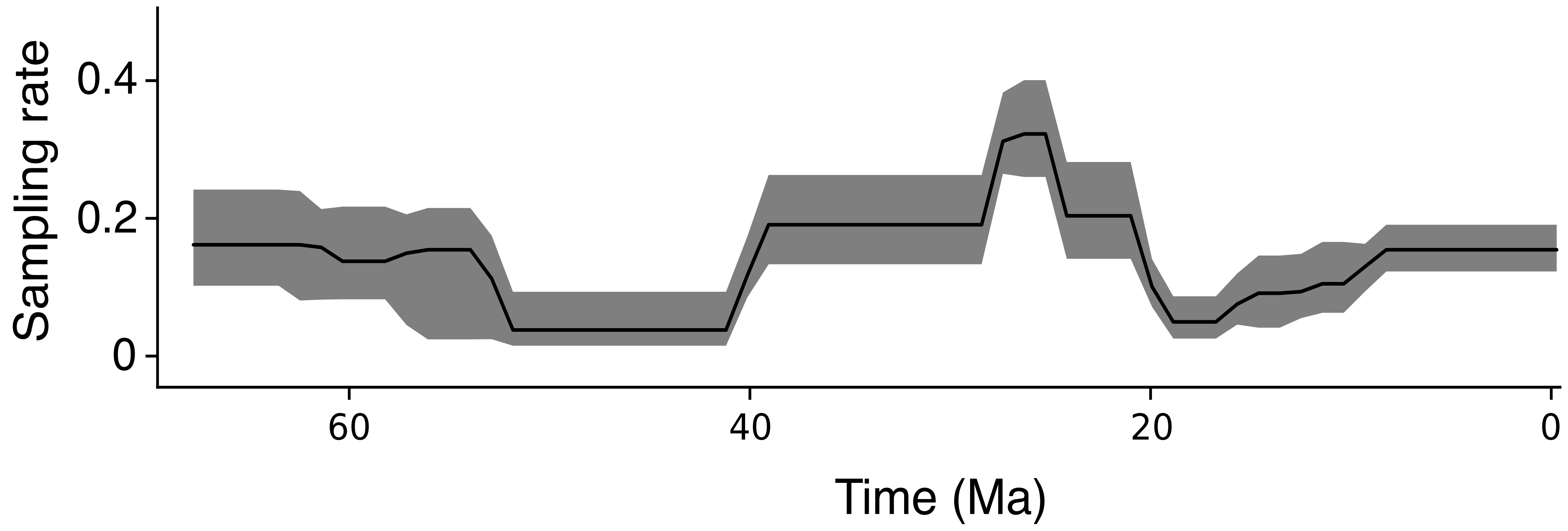
github.com/dsilvestro/PyRate

Silvestro et al. 2019 Paleobiology

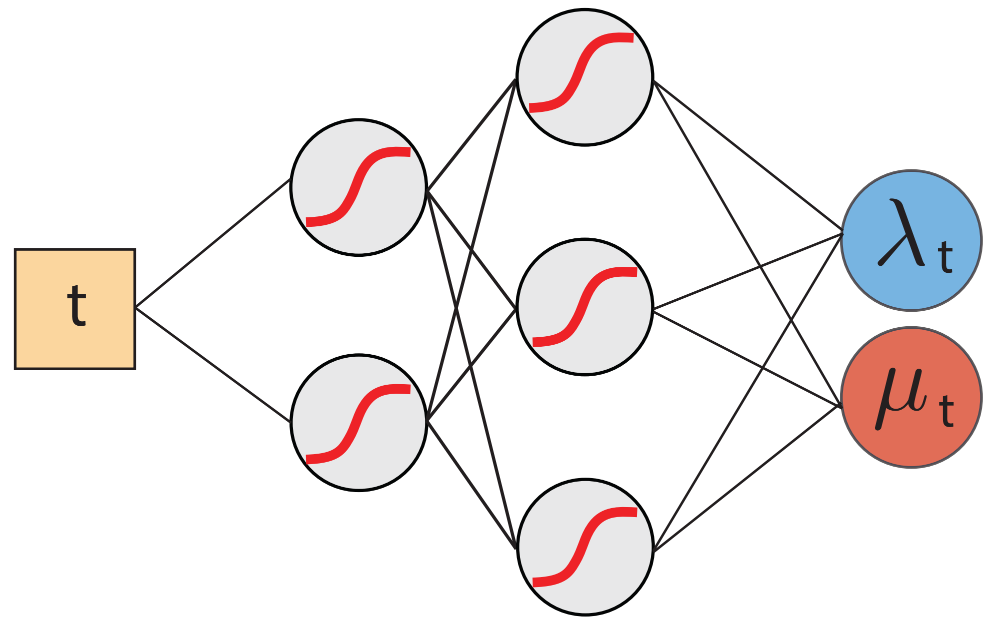
A birth-death neural network model of speciation and extinction



Preservation rates



Time-varying **Speciation** & **extinction** rates modeled by a NN



github.com/dsilvestro/PyRate

Hauffe et al. 2024 Science Advances