“Deep learning from scratch”

~ Chapter 5 *“Back propagation”* ~

Chapter 5.1 Computational graph
Chapter 5.2 Chain rule
Chapter 5.3 backward propagation

2018/06/18

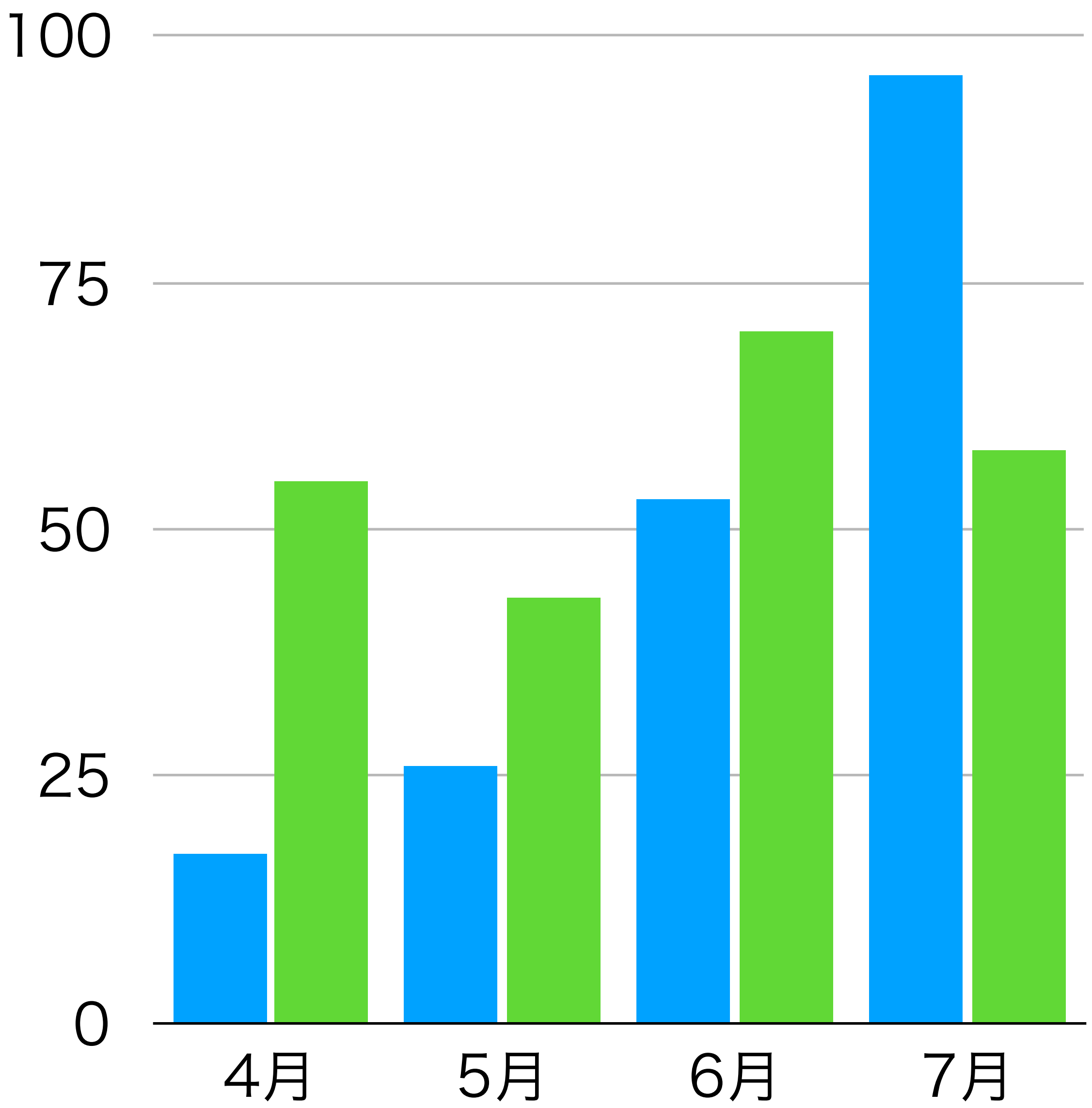
Yousuke Ogata

5.1 Computational Graph

- To understand backpropagation, explain by…
  - Mathematics
    -> Strict, but tooooooo hard to learn ***all***
  - Computational graph
    -> could be understood visually: more easier…?

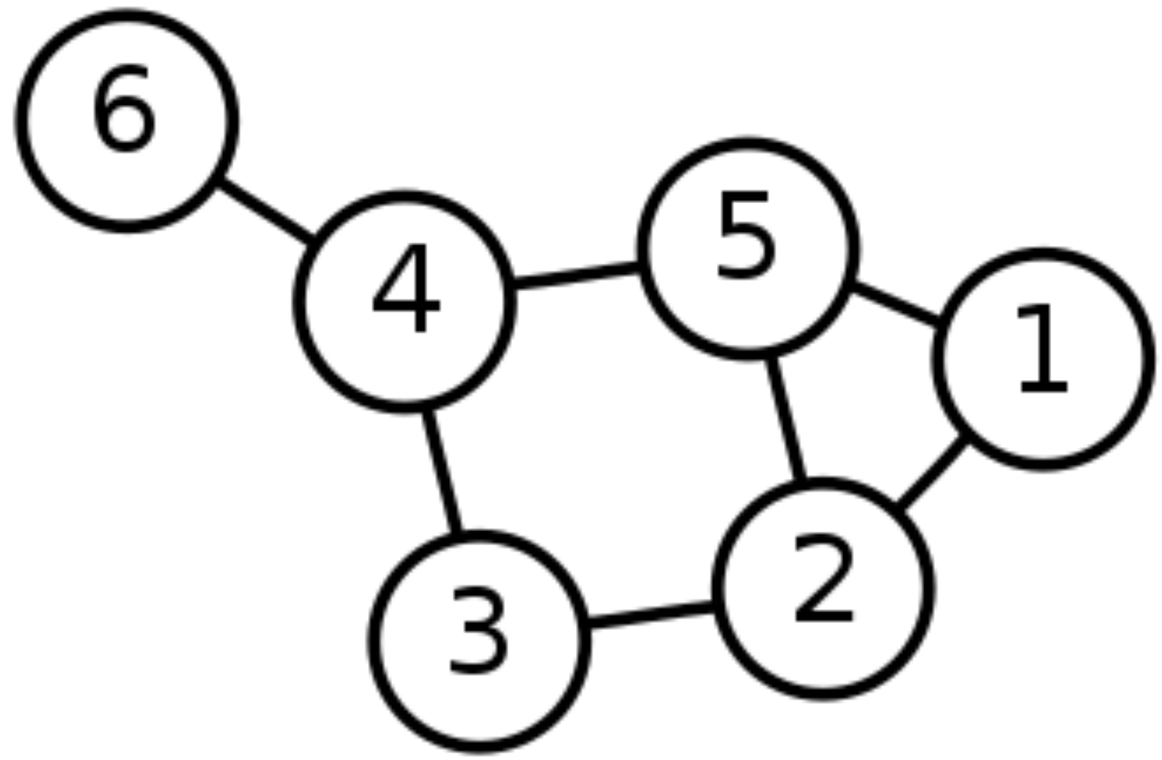
    ref) lecture in Stanford univ, “CS231n”
    (http://cs231n.github.io/)

What is the “Graph?”



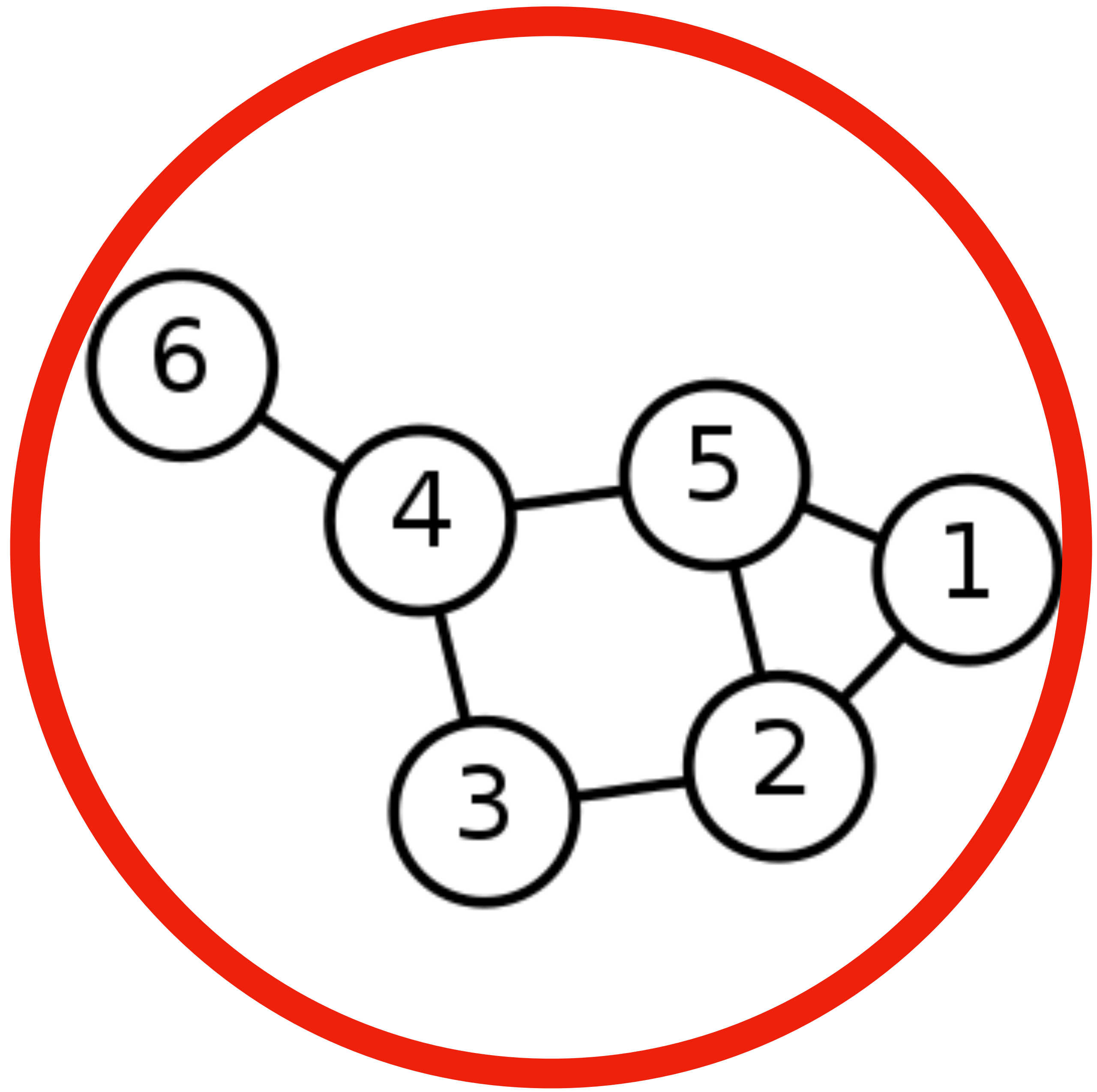
What is the “Graph?”

What is the “Graph?”



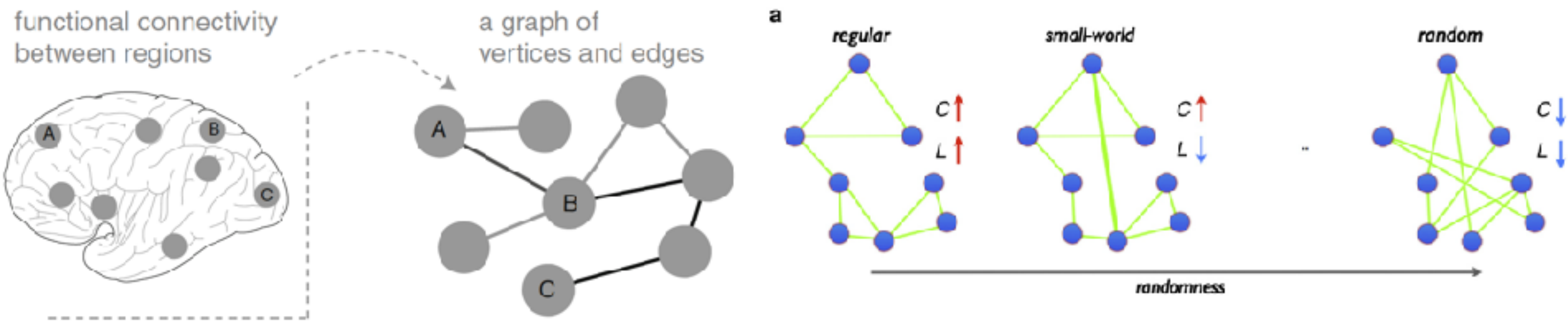
(Wikipedia: “Graph theory”
https://en.wikipedia.org/wiki/Graph_theory , 2018/06/15)

What is the “Graph?”



(Wikipedia: “Graph theory”
https://en.wikipedia.org/wiki/Graph_theory , 2018/06/15)

Graph theory (in Neuroimaging)

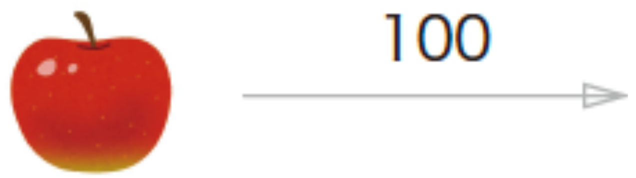


- Construct from Vertices (Nodes) and Edges
  - Vertex : ROI or single-voxel
  - Edge : functional connectivity
    - Path :a sequence of vertices in which all succeeding vertices are connected by edges
- To analyze relationship of graph, calculate
  - Distance : the minimum length among all paths connecting vertices
  - Degree : the number of edges connecting to it

Cf) Seven Bridges of Königsberg, four-color problem etc…

Solving problems with computational graph

- Q1: Taro bought two apples. Apple is priced ¥100 (with exclude VAT:10%).
  How much taro needs to pay?



Solving problems with computational graph

- Q1: Taro bought two apples. Apple is priced ¥100 (with exclude VAT:10%).
  How much taro needs to pay?

Solving problems with computational graph

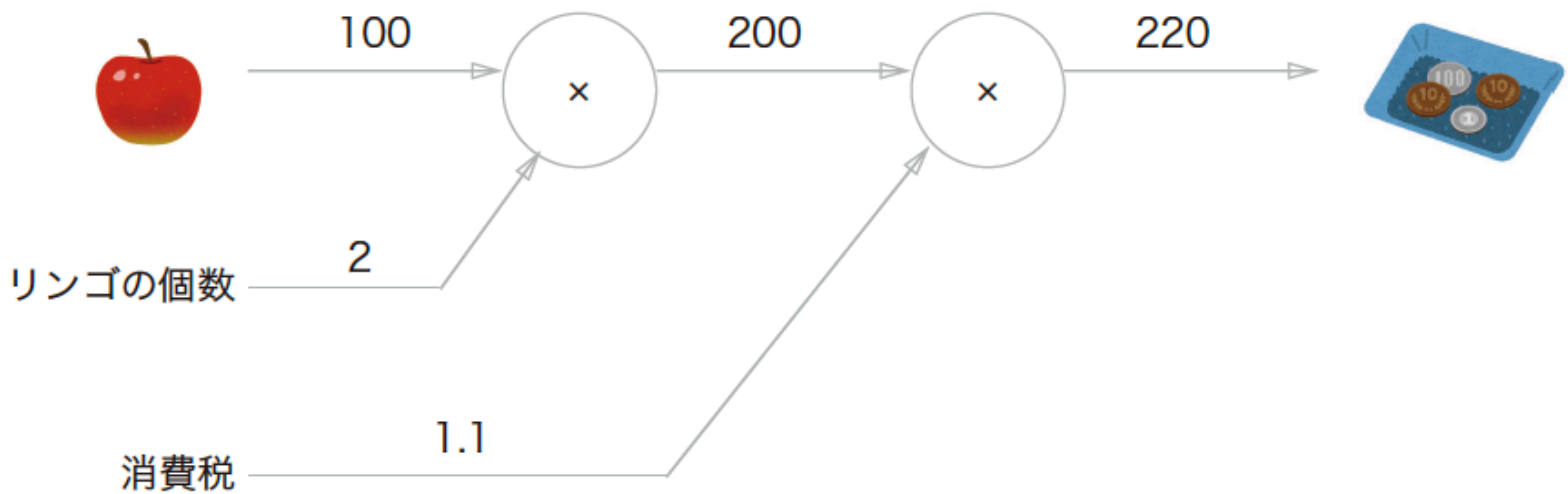
- Q1: Taro bought two apples. Apple is priced ¥100 (with exclude VAT:10%).
  How much taro needs to pay?

Solving problems with computational graph

- Q1: Taro bought two apples. Apple is priced ¥100 (with exclude VAT:10%).
  How much taro needs to pay?

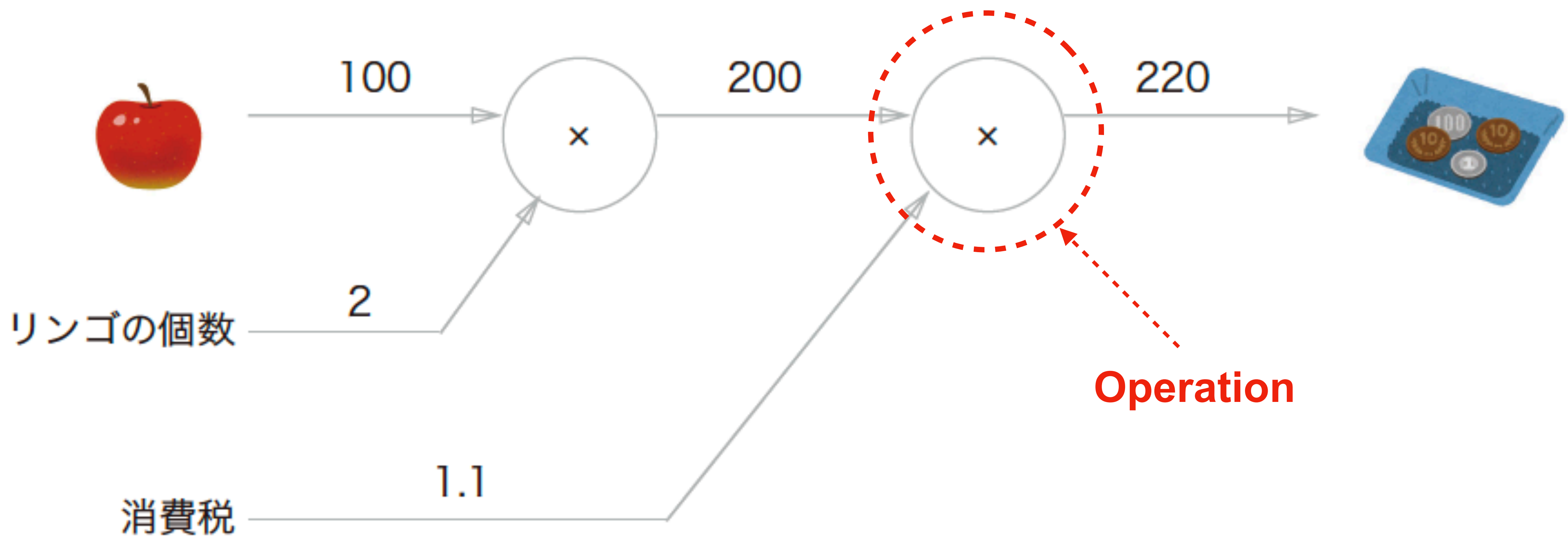
Solving problems with computational graph

- Q1: Taro bought two apples. Apple is priced ¥100 (with exclude VAT:10%).
  How much taro needs to pay?



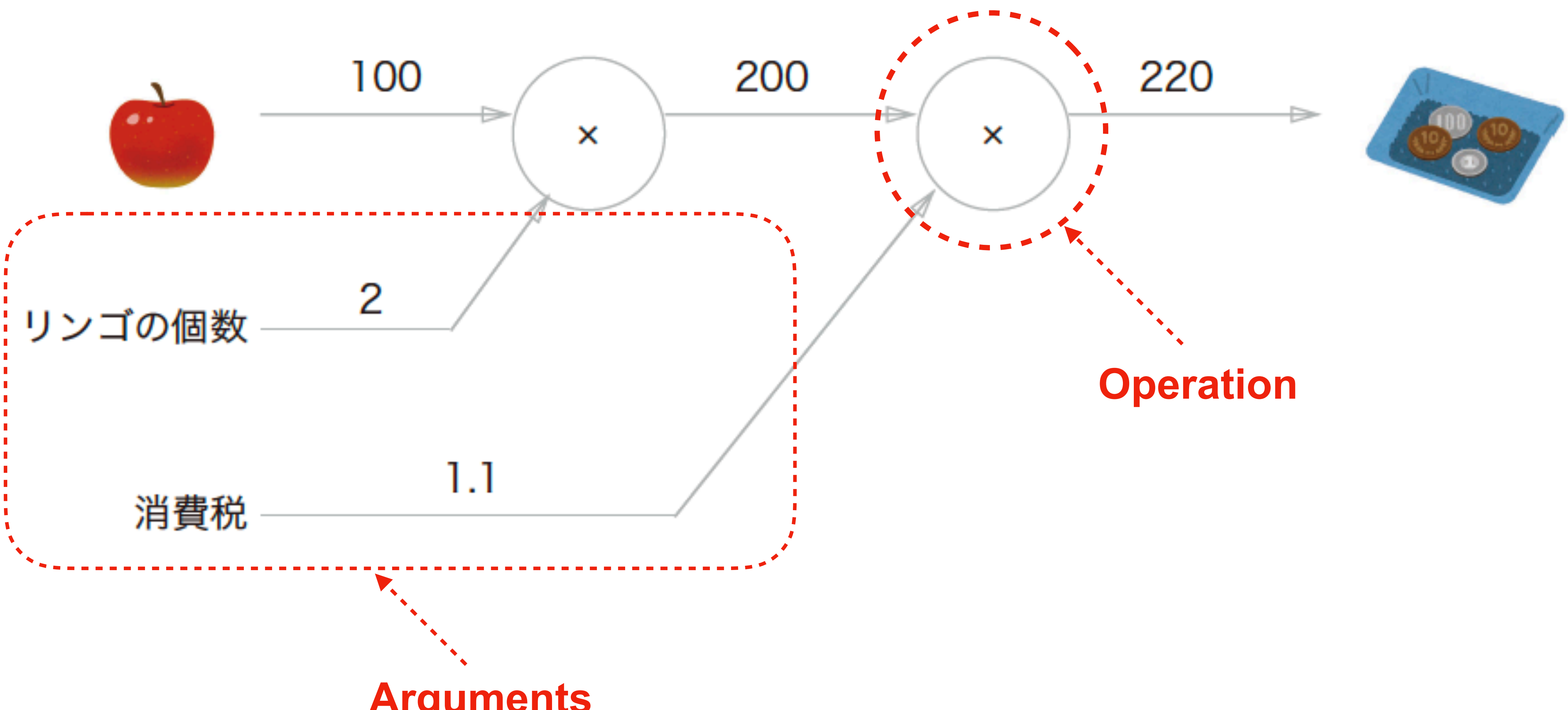
Solving problems with computational graph

- Q1: Taro bought two apples. Apple is priced ¥100 (with exclude VAT:10%).
  How much taro needs to pay?



Solving problems with computational graph

- Q1: Taro bought two apples. Apple is priced ¥100 (with exclude VAT:10%).
  How much taro needs to pay?

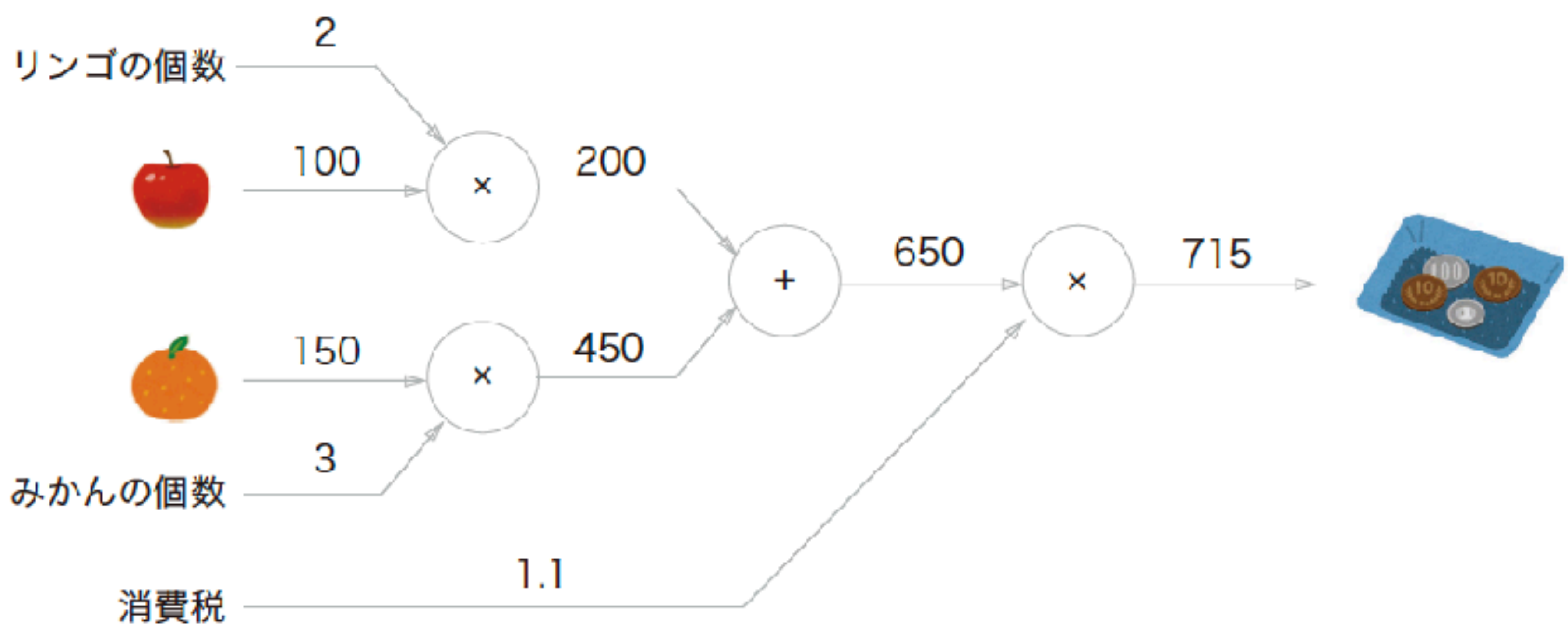


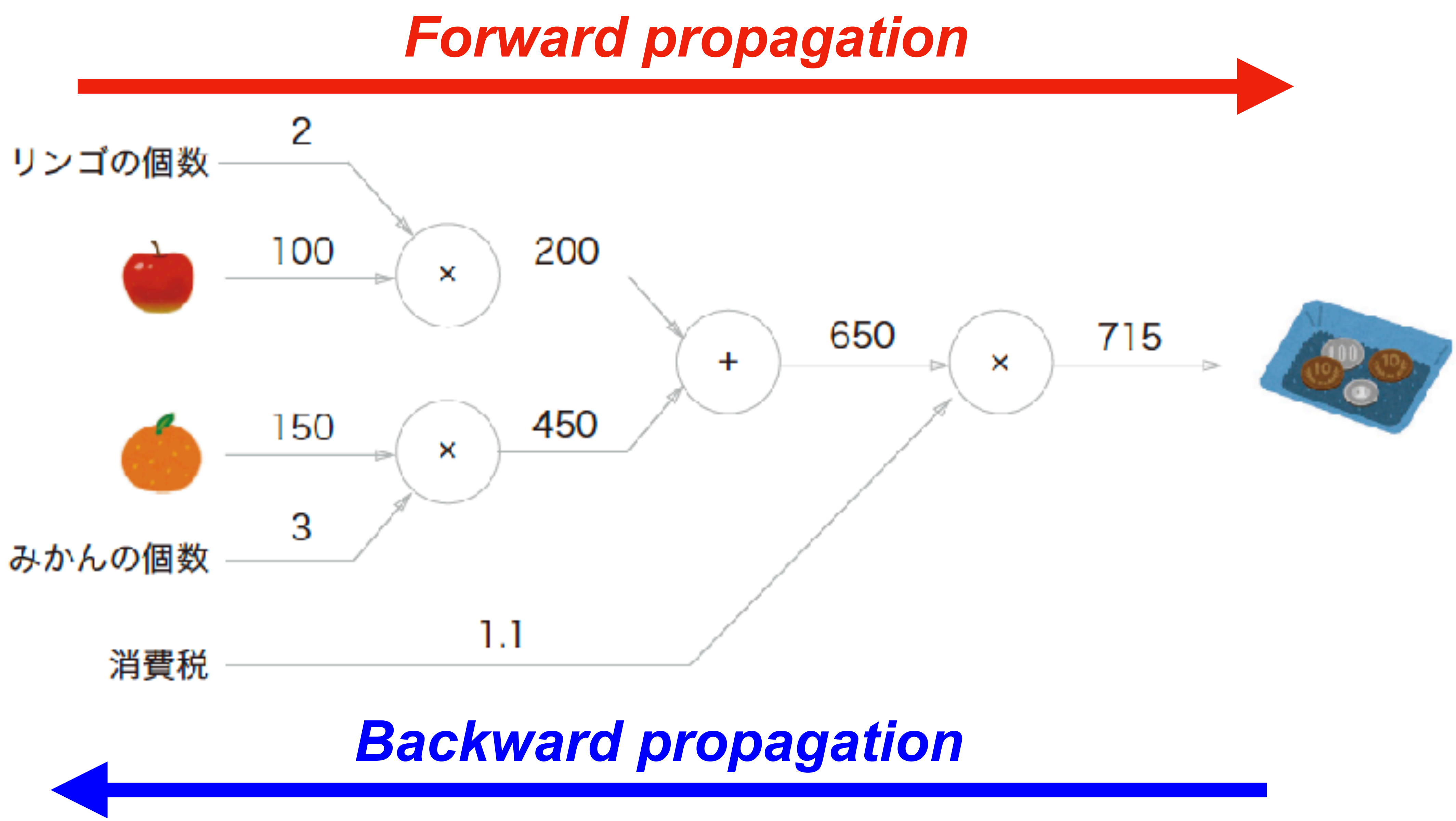
Solving problems with computational graph (2)

- Q2: Taro bought two apples and three oranges.
  Apple is priced ¥100, Orange is priced ¥150(VAT:10%).
  How much taro needs to pay?

Solving problems with computational graph (2)

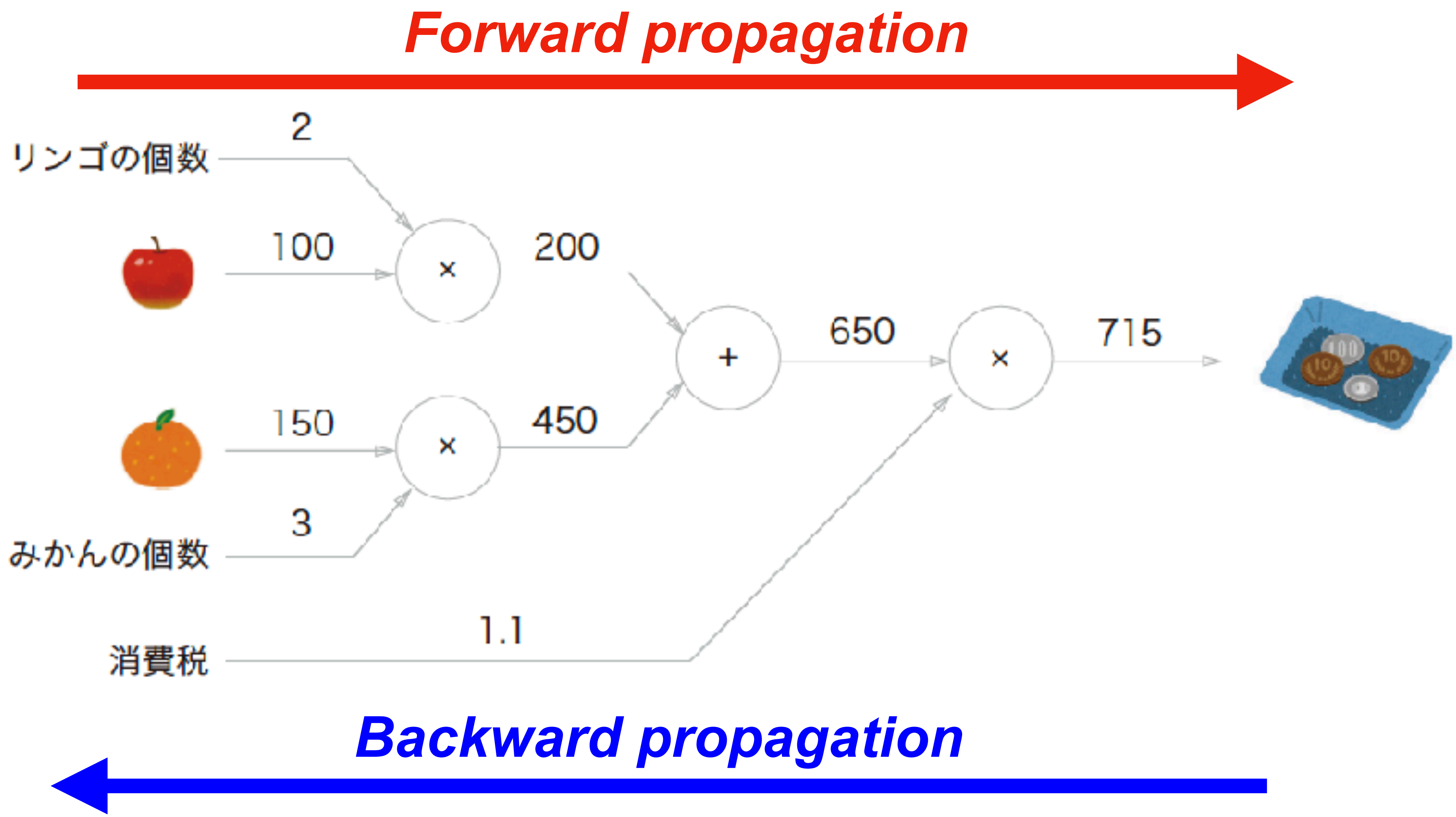
- Q2: Taro bought two apples and three oranges.
  Apple is priced ¥100, Orange is priced ¥150(VAT:10%).
  How much taro needs to pay?





Forward propagation
リンゴの個数
2
100
200
150
450
3
みかんの個数
650
715
消費税
1.1
Backward propagation

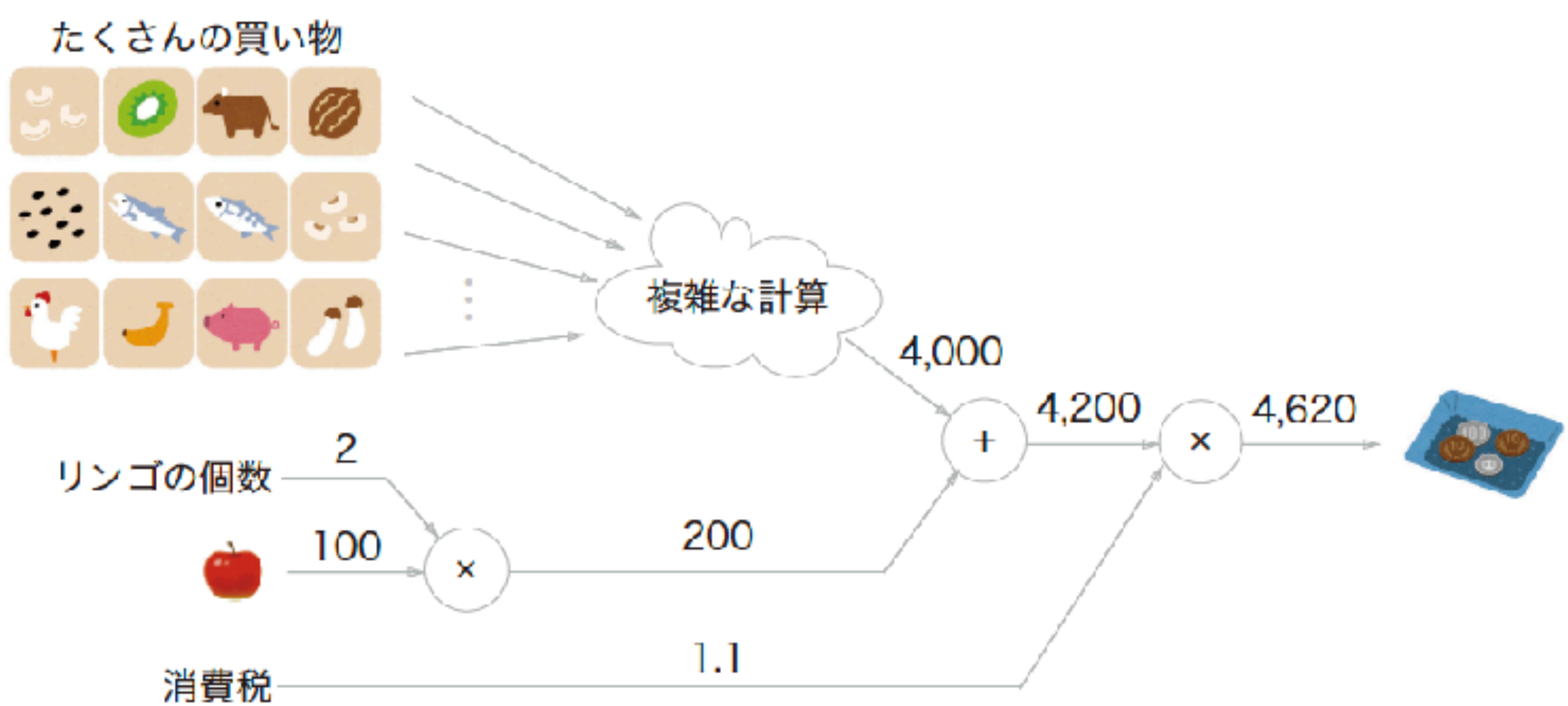
computes values from inputs(left) to output(right)
=> ***forward propagation***



transfer gradient from output(right) to inputs(left)
=> ***Backward propagation***

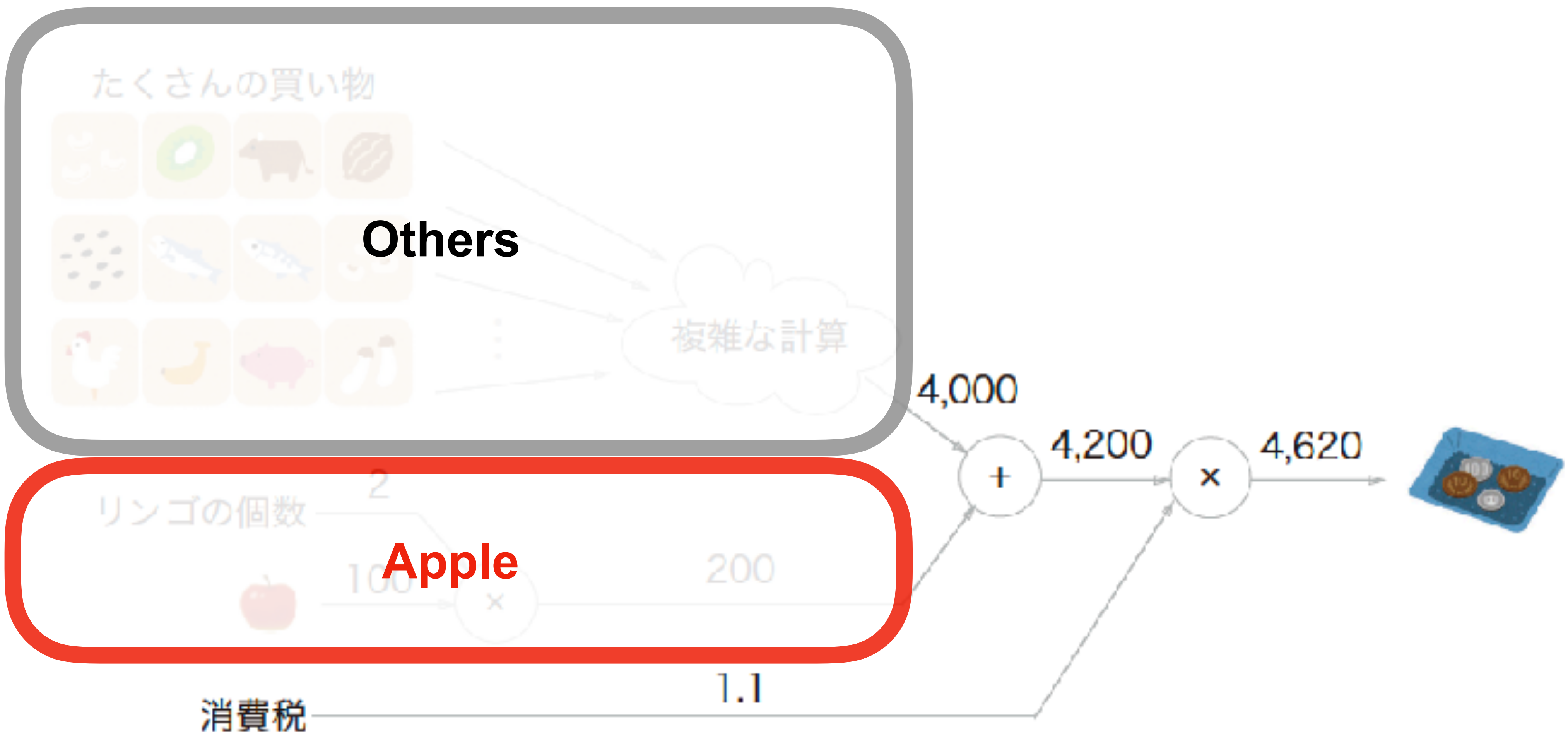
“Local” processing

- Computational graph allowed us to obtain a result by transferring “local operations”
  => can ignore “global” processing



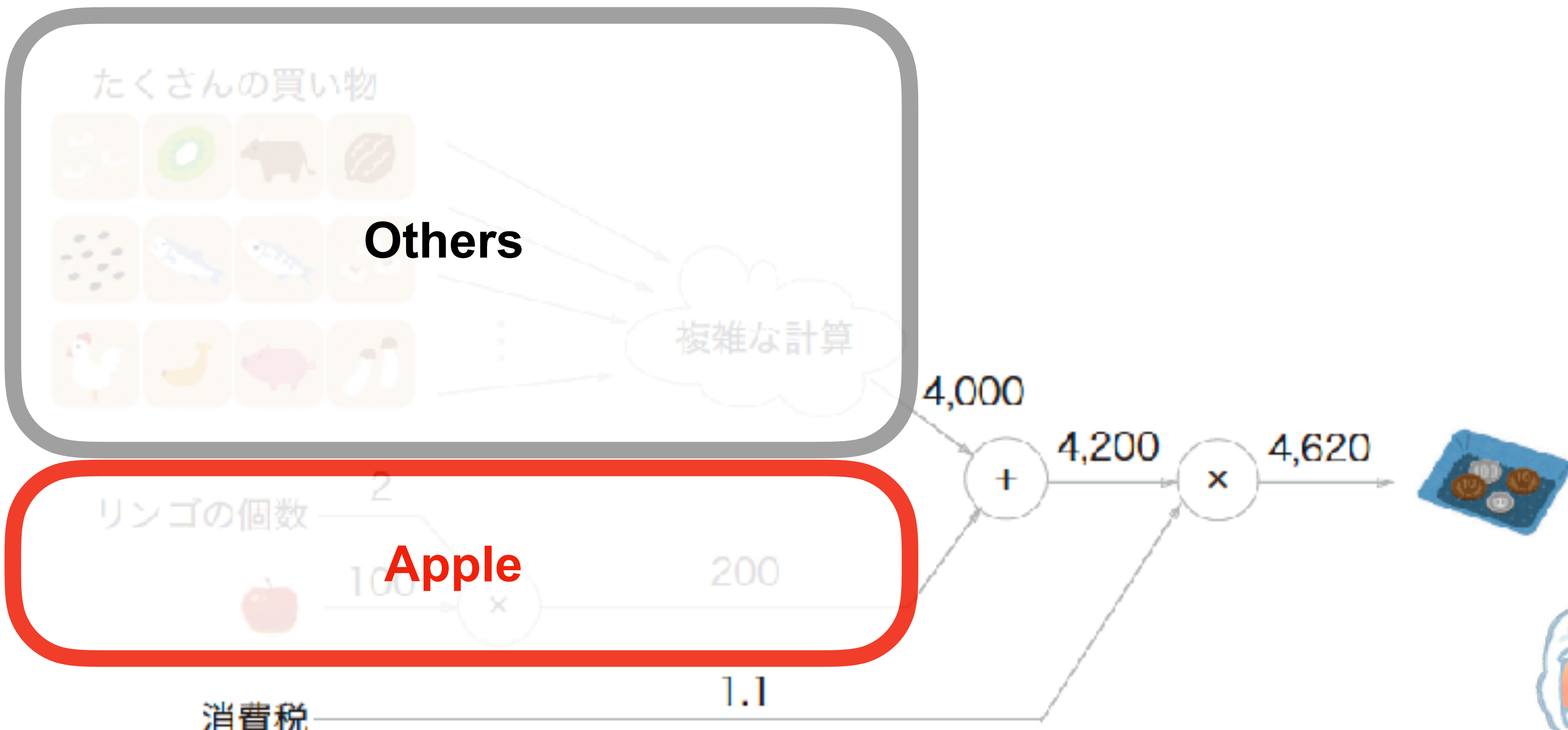
“Local” processing

- Computational graph allowed us to obtain a result by transferring “local operations”
  => can ignore “global” processing



“Local” processing

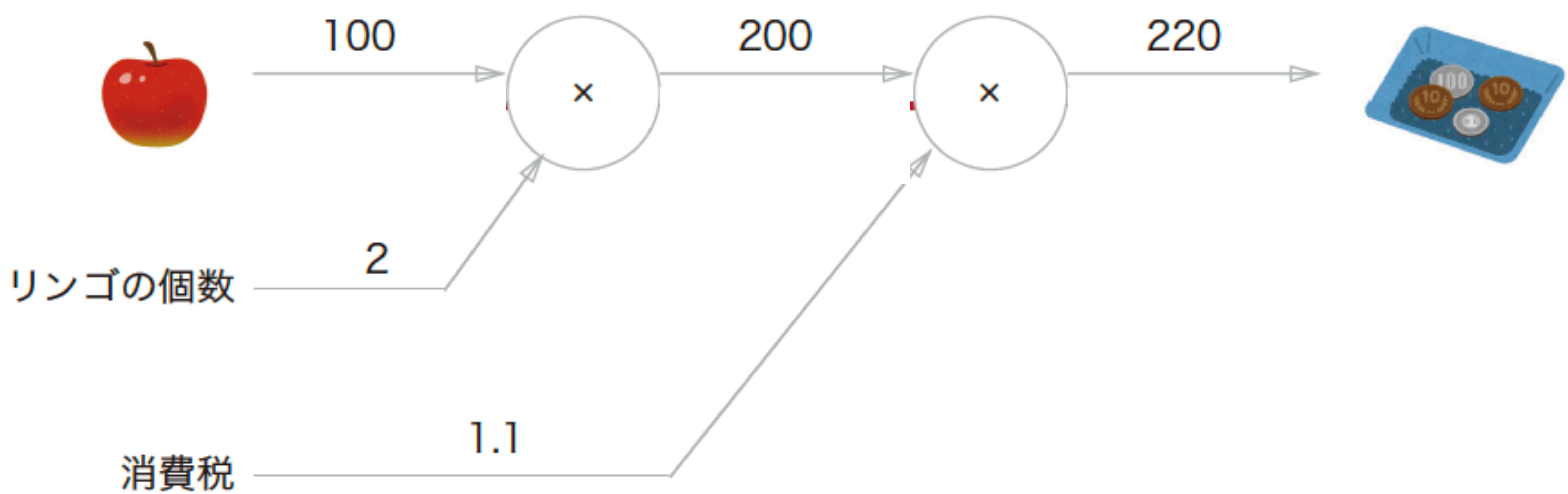
- Computational graph allowed us to obtain a result by transferring “local operations”
  => can ignore “global” processing



Just means…
= can focus only “local” operations in each nodes

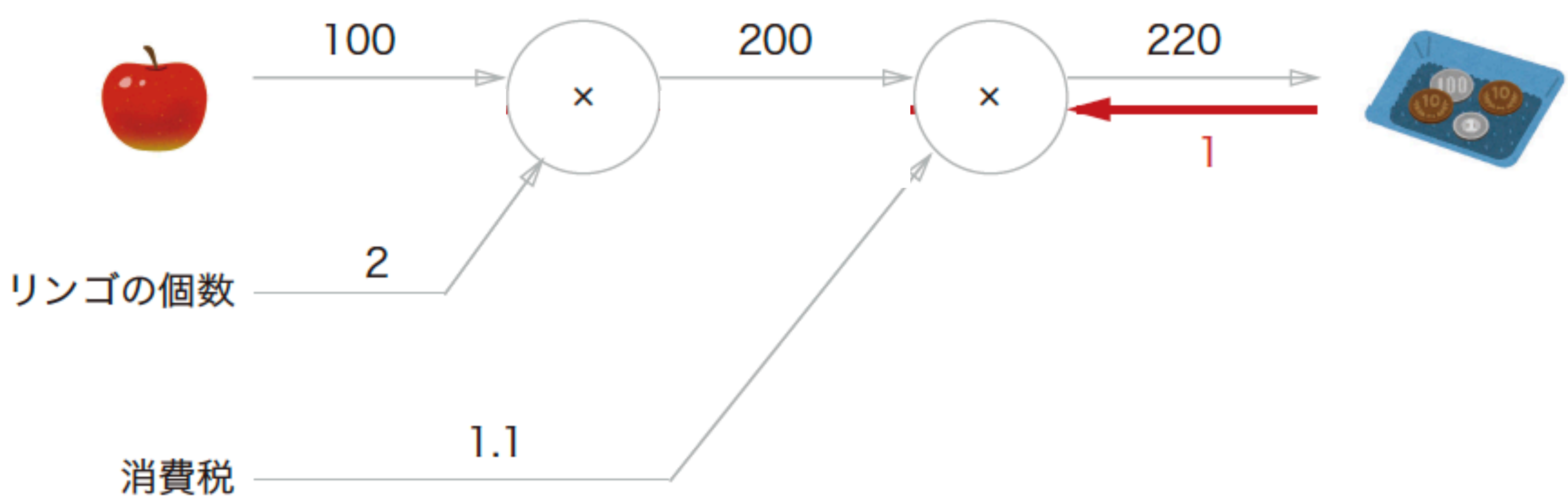
Backward propagation

- Why computational graph was used for explain backward propagation??
  => calculate gradient efficiently



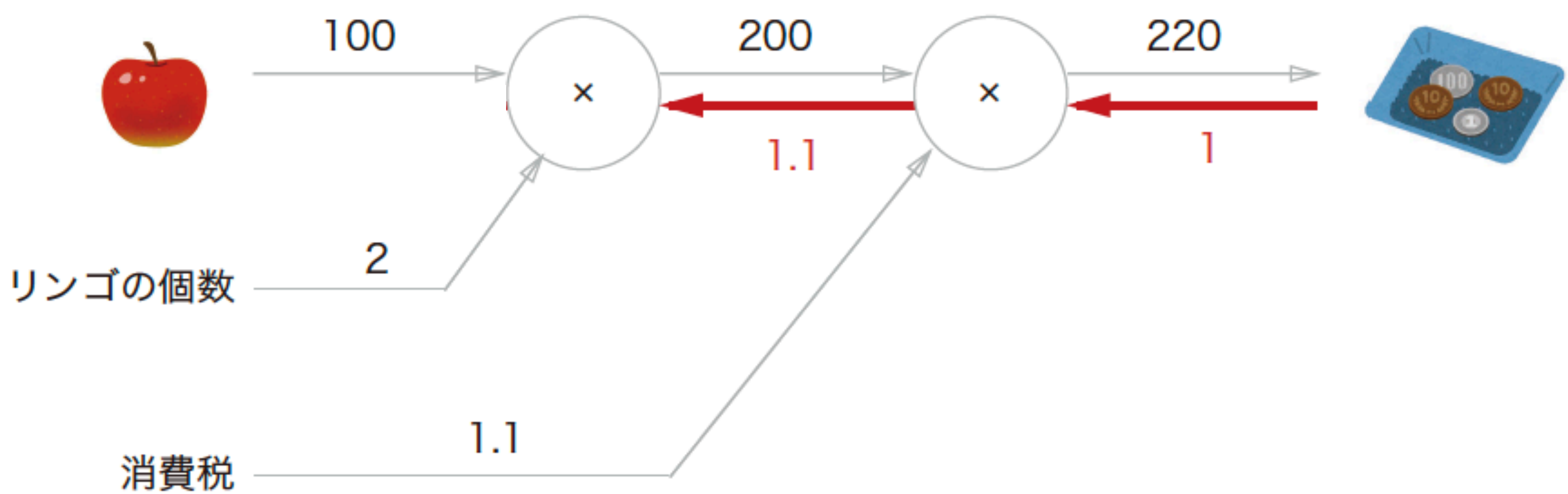
Backward propagation

- Why computational graph was used for explain backward propagation??
  => calculate gradient efficiently



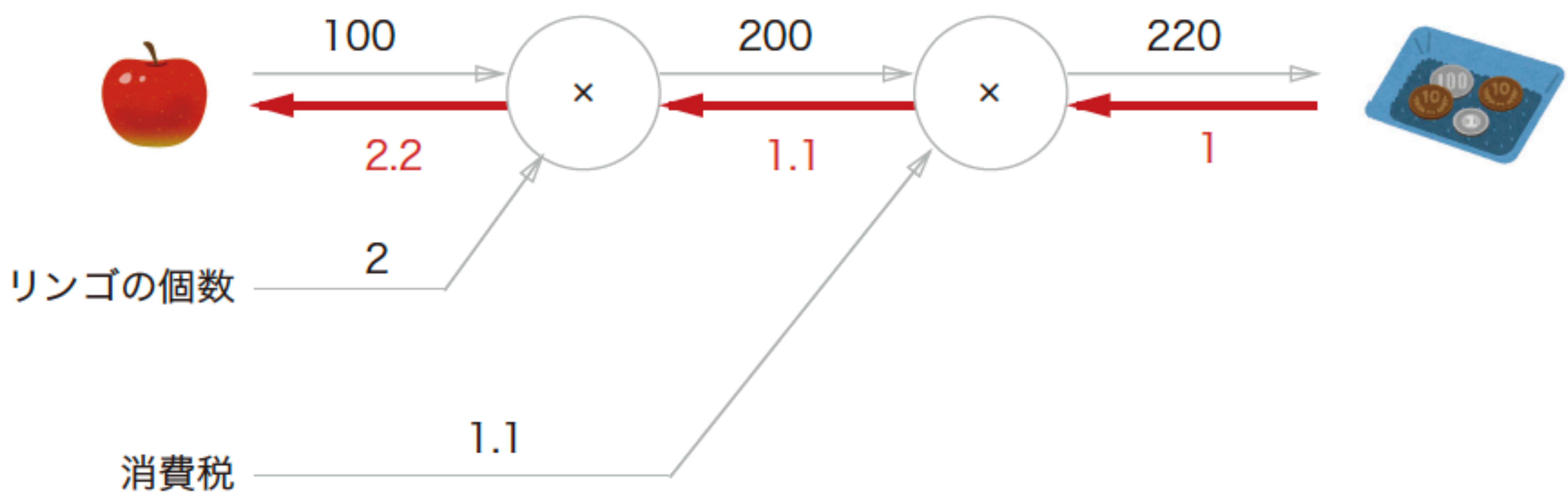
Backward propagation

- Why computational graph was used for explain backward propagation??
  => calculate gradient efficiently



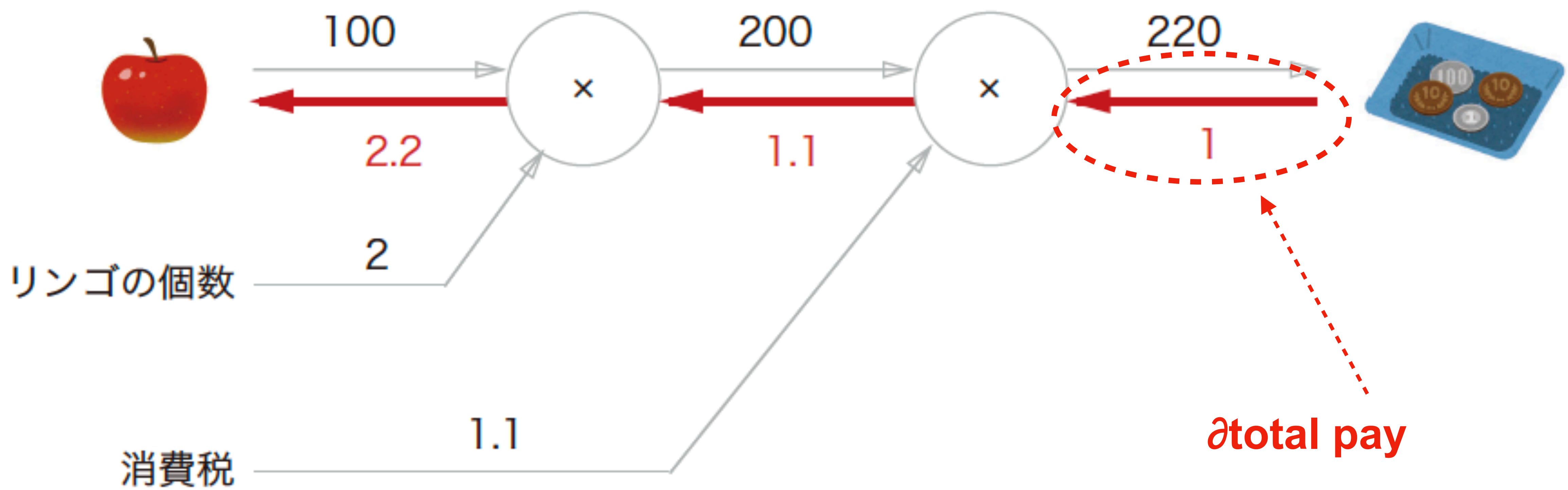
Backward propagation

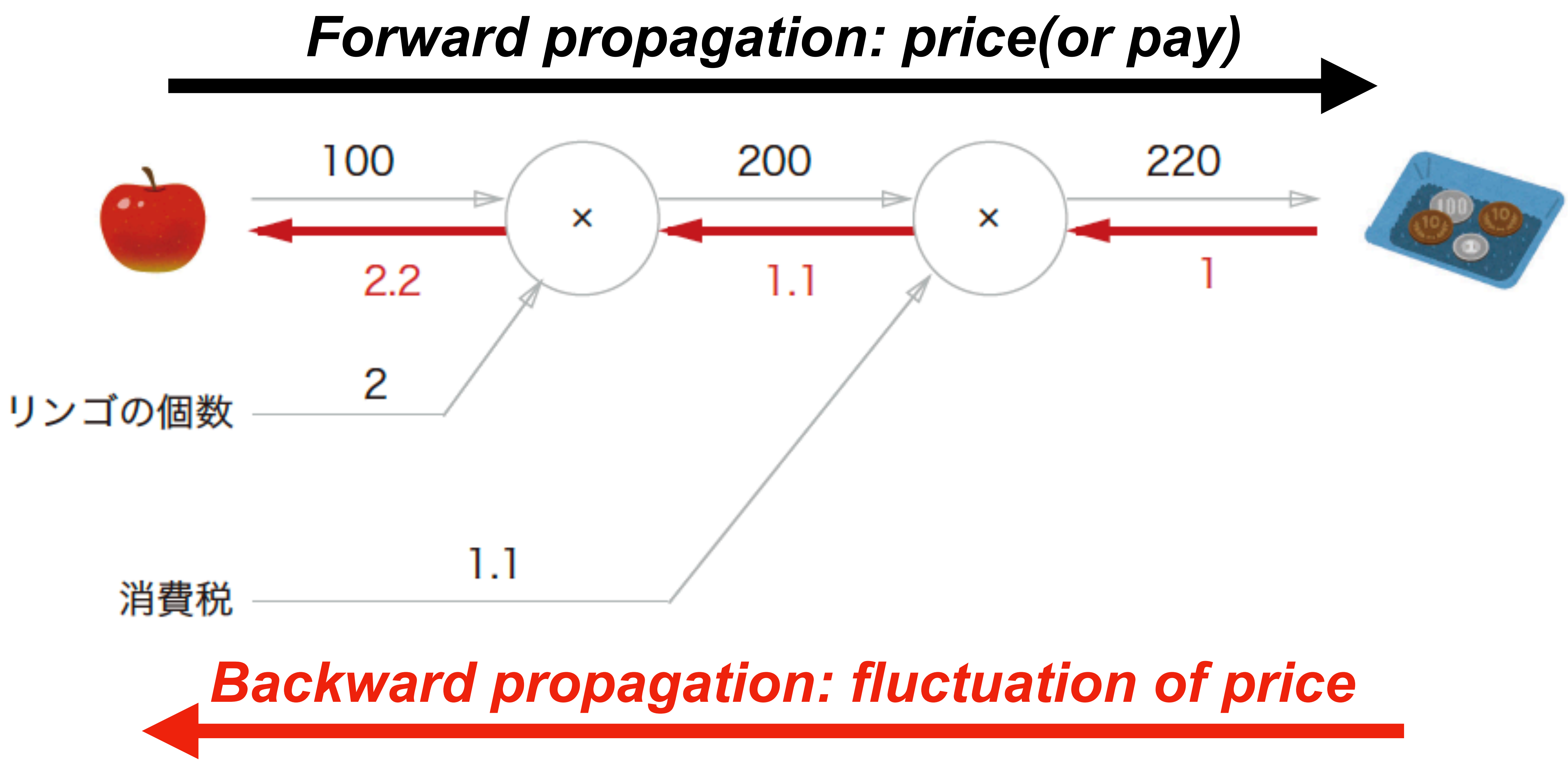
- Why computational graph was used for explain backward propagation??
  => calculate gradient efficiently



Backward propagation

- Why computational graph was used for explain backward propagation??
  => calculate gradient efficiently





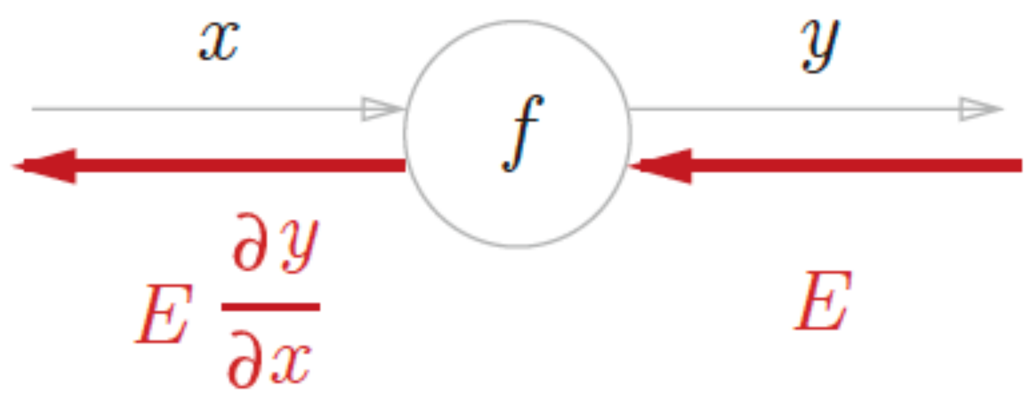
Forward propagation: price(or pay)
100
×
200
×
220
2.2
1.1
1
リンゴの個数
2
消費税
1.1
Backward propagation: fluctuation of price

5.2 Chain rule

- Chain rule: a formula for computing the *derivative* of the *composition of two or more functions*.
  (from wikipedia, “Chain rule”)

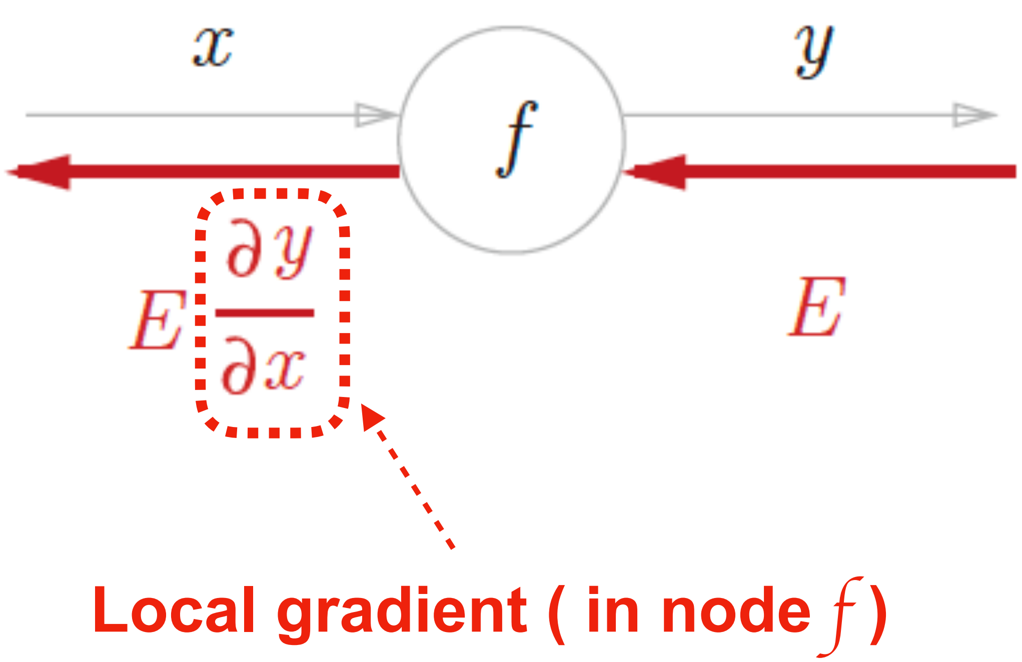
5.2 Chain rule

- In backpropagation, it pass “local” gradient to previous node.
  => based on chain rule



5.2 Chain rule

- In backpropagation, it pass “local” gradient to previous node.
  => based on chain rule



Composite function

- Composite function: function composed of multiple functions

$$z = (x+y)^2$$

$$\Downarrow$$

$$\begin{aligned} z &= t^2 \\ t &= x + y \end{aligned}$$

Chain rule:
When a function is represented by a composite function, the derivative of the composite function can be represented by the product of the differentiation of the each functions.

Chain rule:
When a function is represented by a composite function, the derivative of the composite function can be represented by the product of the differentiation of the each functions.

$$\begin{aligned} z &= t^2 \\ t &= x + y \end{aligned} \quad \Longrightarrow \quad \frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\,\frac{\partial t}{\partial x}$$

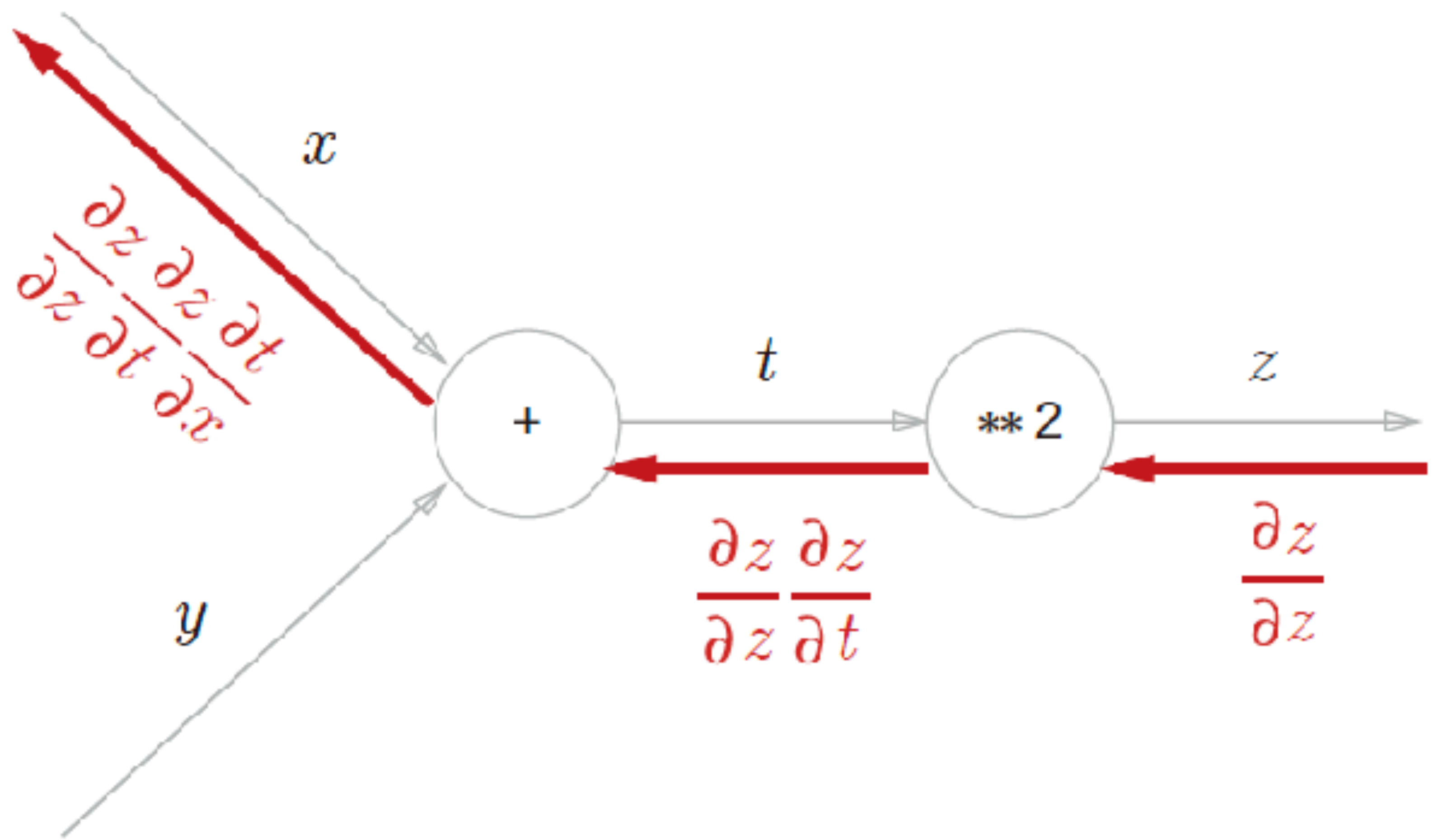
Example:

$$\begin{aligned} \frac{\partial z}{\partial t} &= 2t \\ \frac{\partial t}{\partial x} &= 1 \end{aligned}$$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\frac{\partial t}{\partial x} = 2t \cdot 1 = 2(x + y)$$

Chain rule in graph

$$
\begin{aligned}
z &= t^2 \\
t &= x + y
\end{aligned}
$$



- In the backpropagation,
  the product of the input to the node and local derivative(= partial derivative) in the node is transferred to next node

5.3 Backpropagation

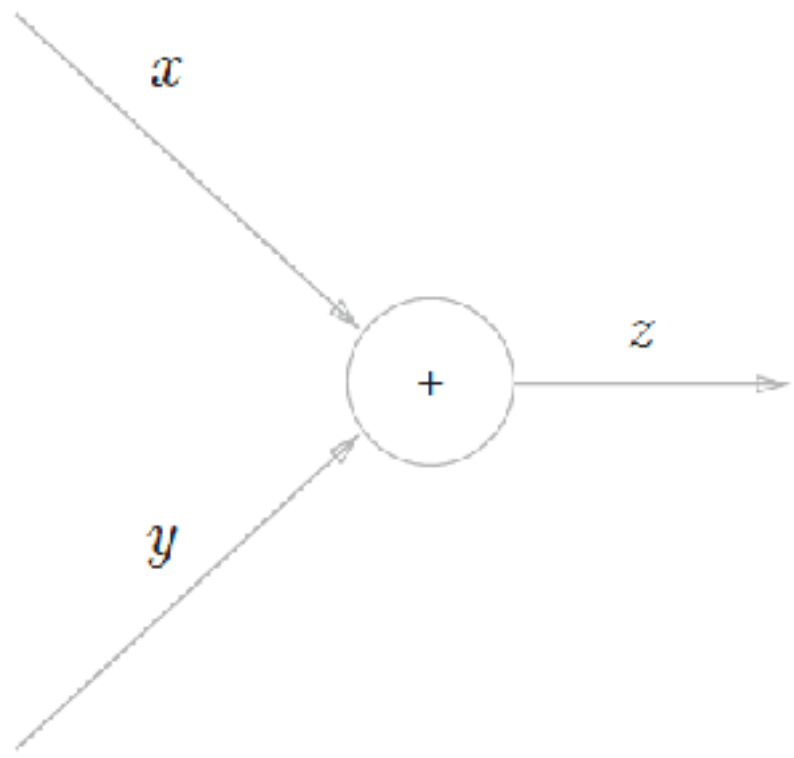
- Backward propagation in ***addition*** node

When

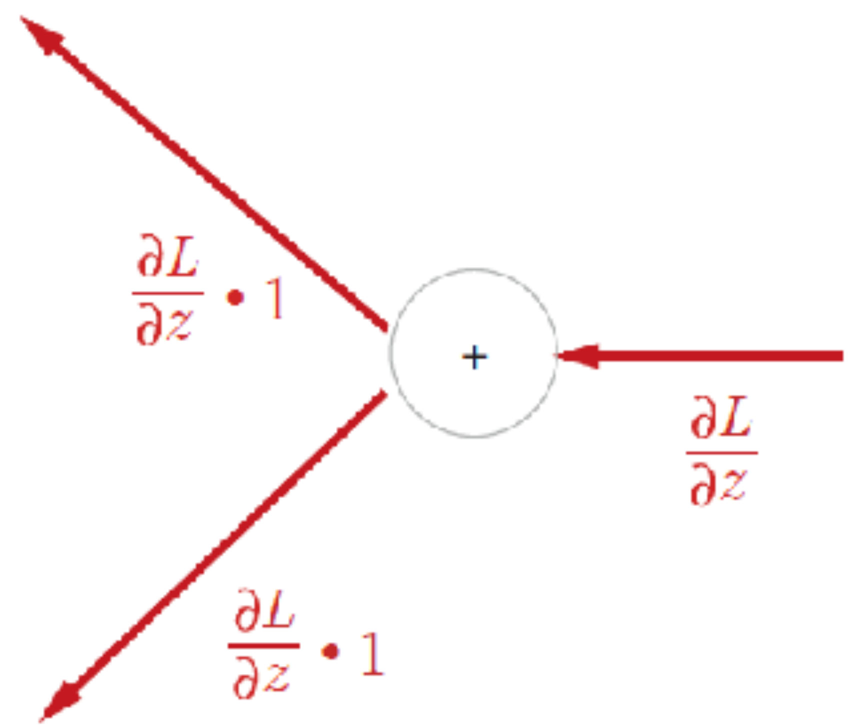
$$z = x + y$$

$$\frac{\partial z}{\partial x} = 1$$

$$\frac{\partial z}{\partial y} = 1$$



Forward propagation



Backward propagation

=> merely transfer input to output as intact

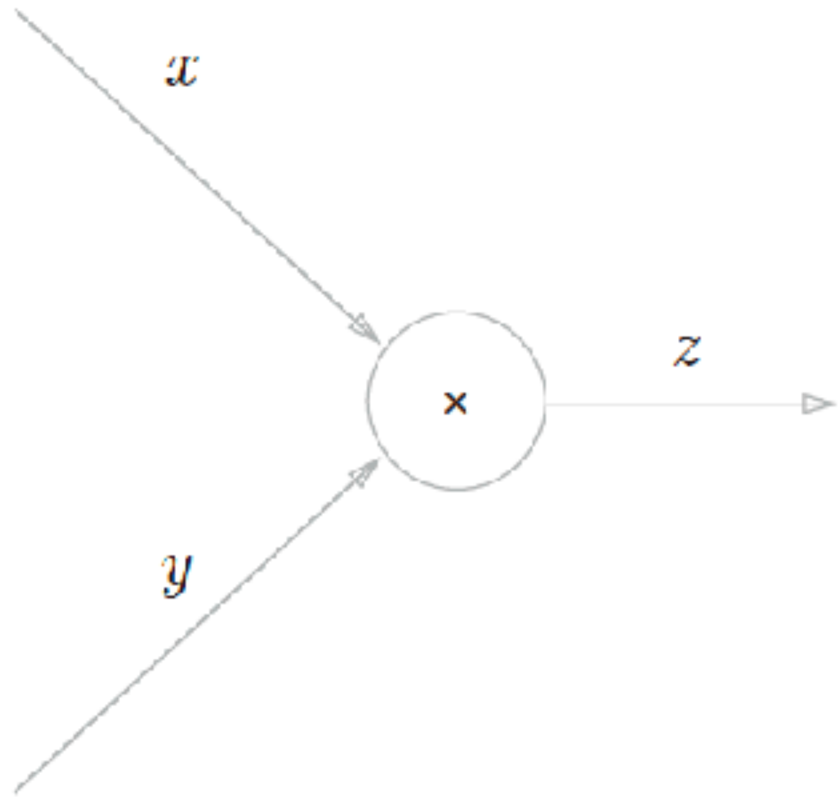
- Backward propagation in ***multiplication*** node

When

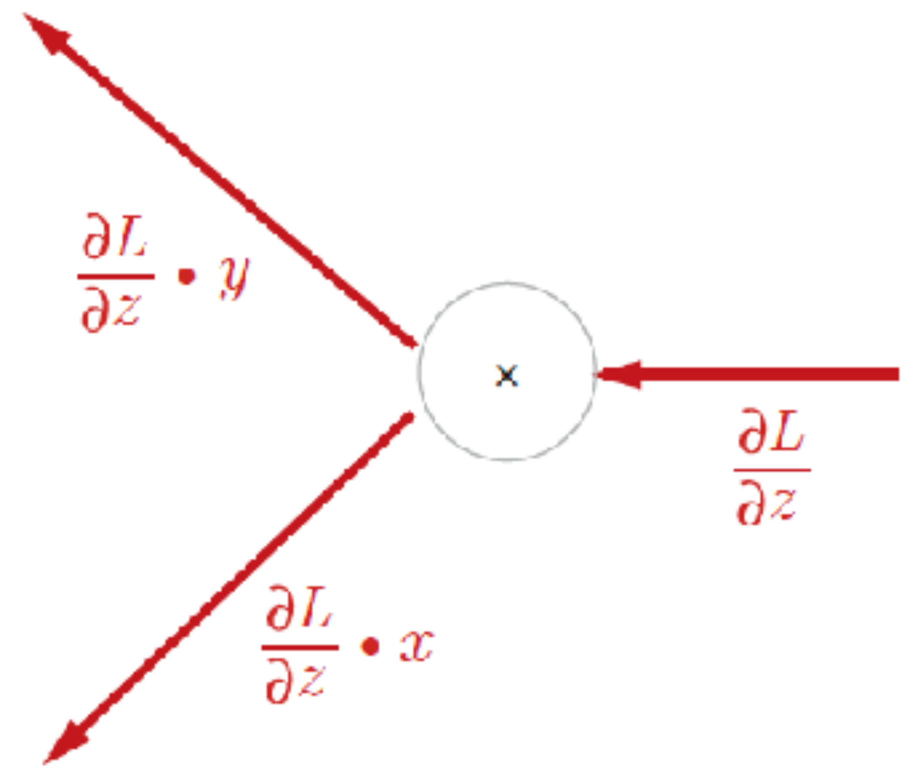
$$z = xy$$

$$\frac{\partial z}{\partial x} = y$$

$$\frac{\partial z}{\partial y} = x$$



Forward propagation

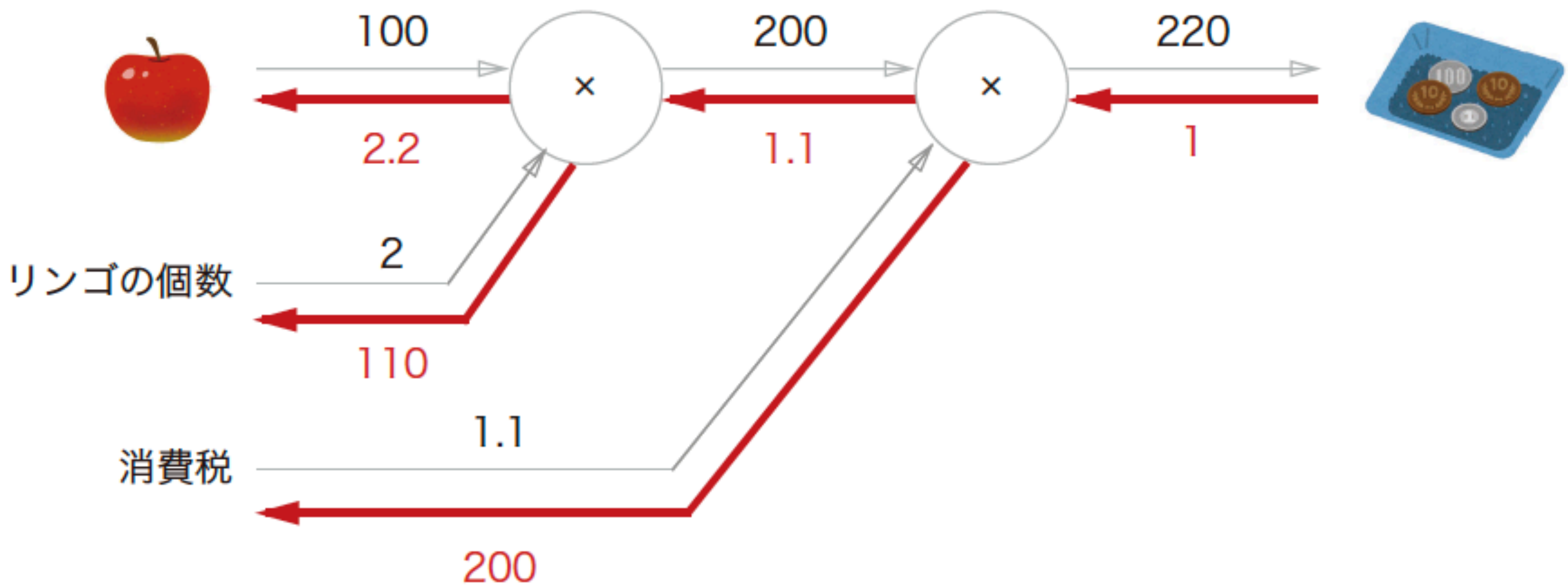


Backward propagation

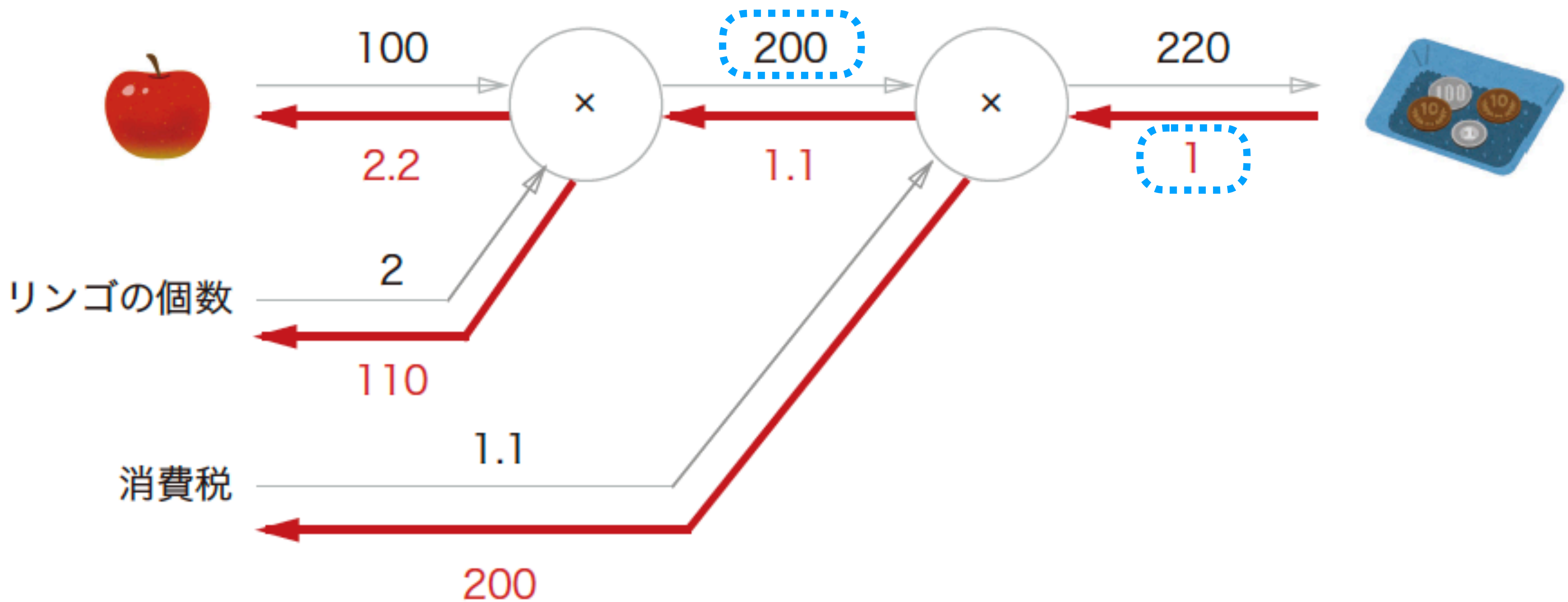
Backpropagation of multiplication needs to the value of input signals
(at forward propagation)

=>Thus, implementation of multiplication node require holding the input value

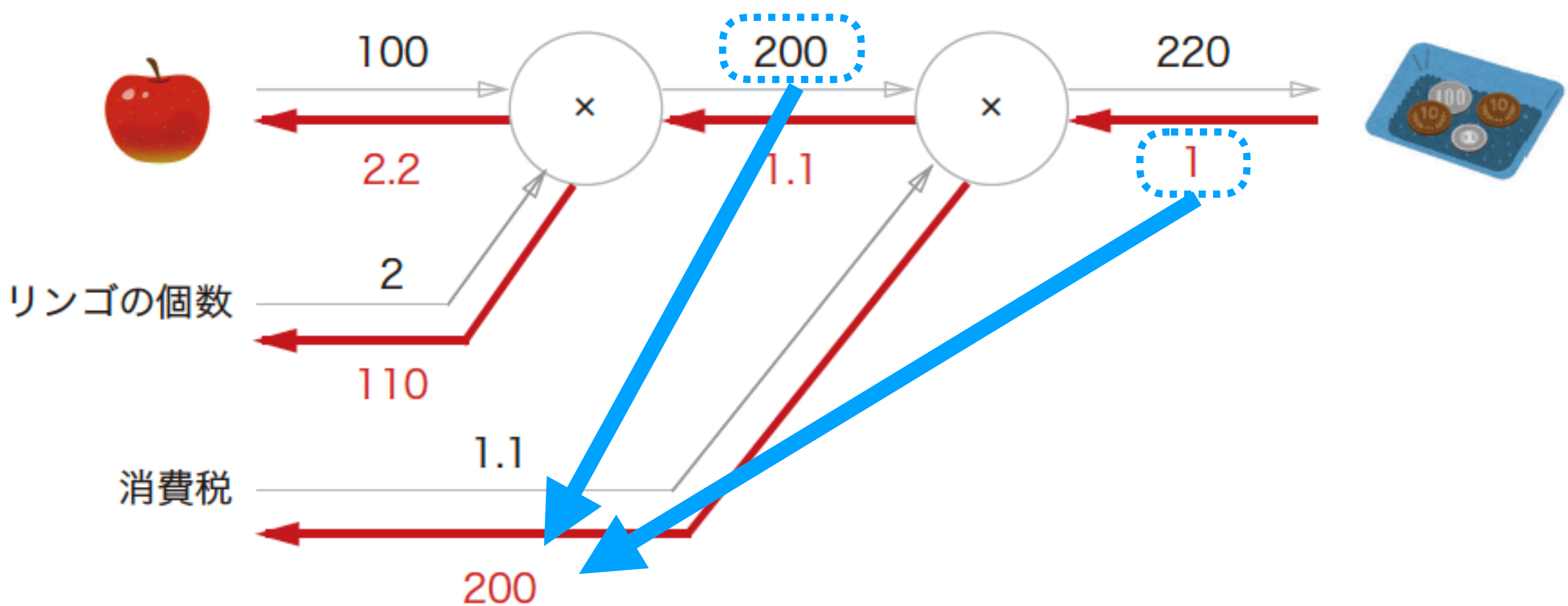
Example: Paid for apple



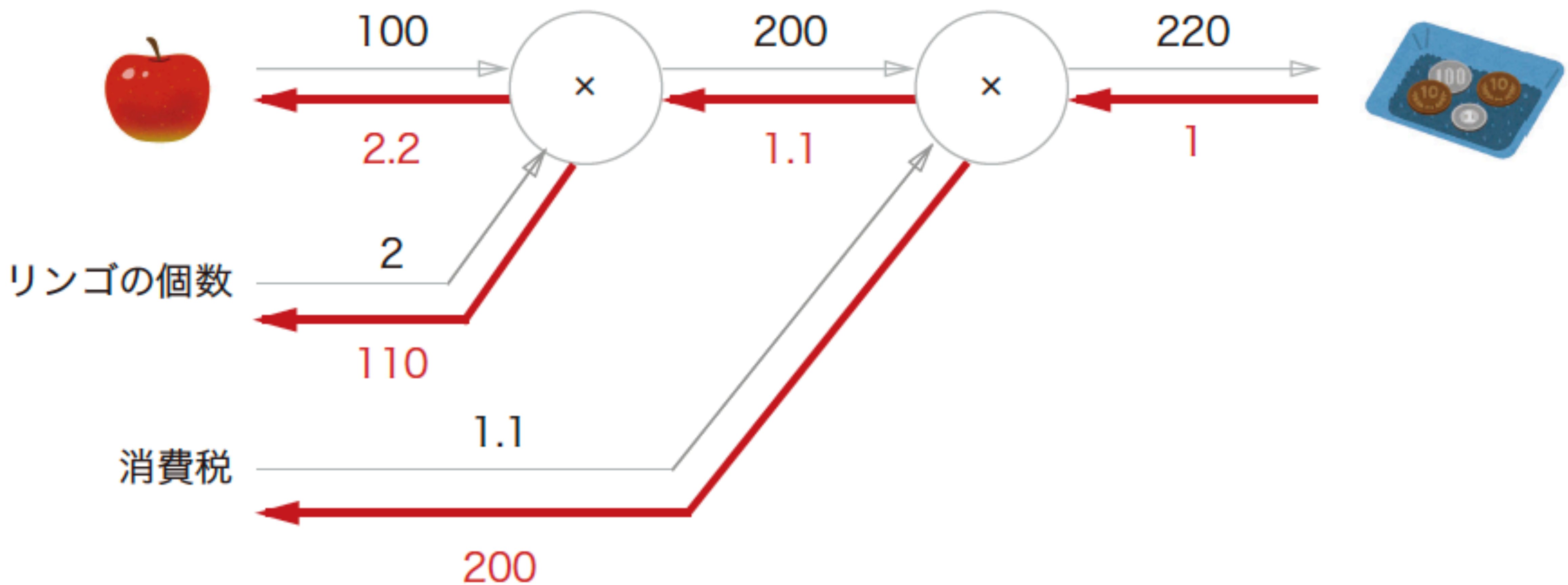
Example: Paid for apple



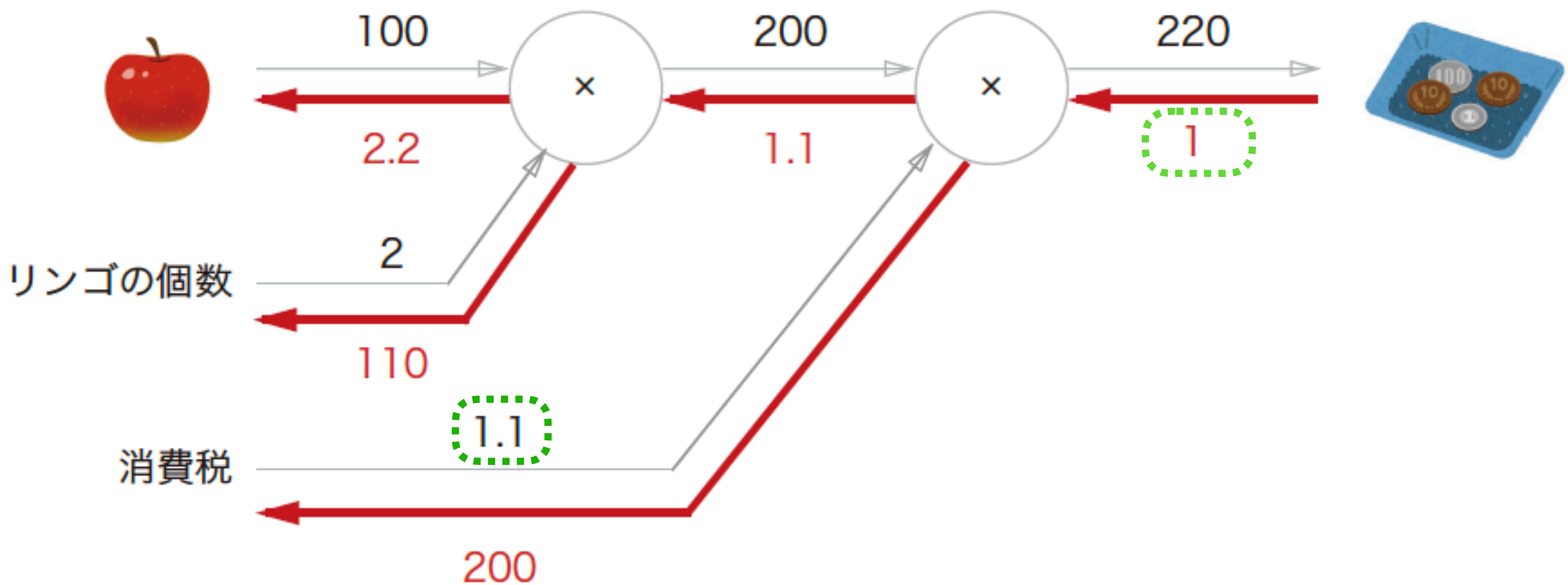
Example: Paid for apple



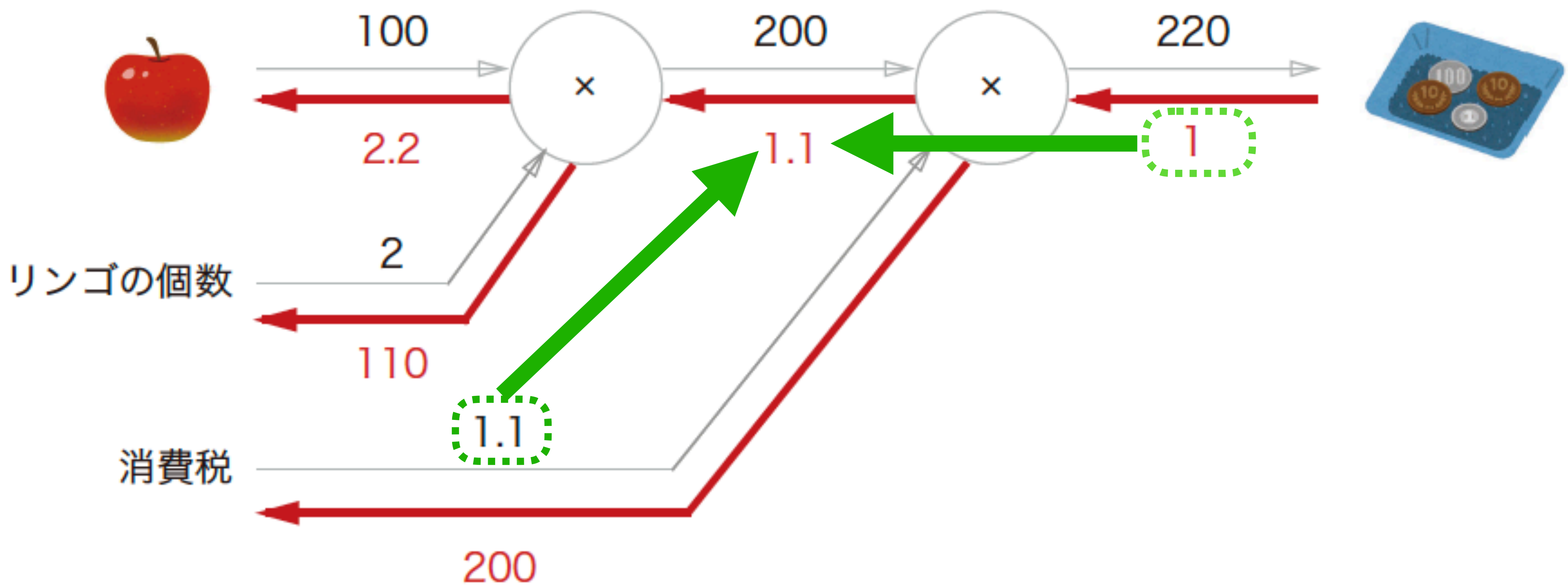
Example: Paid for apple



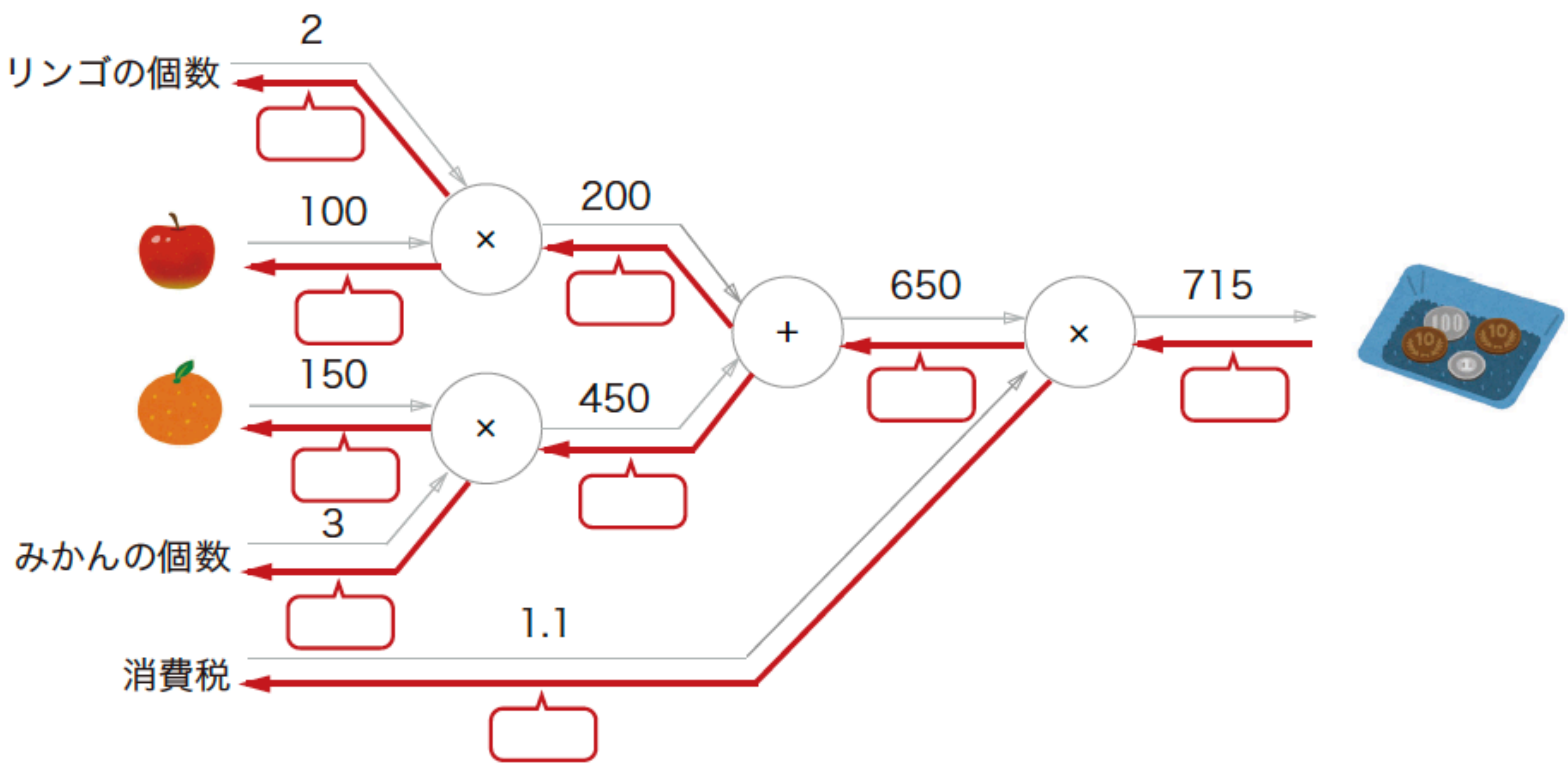
Example: Paid for apple



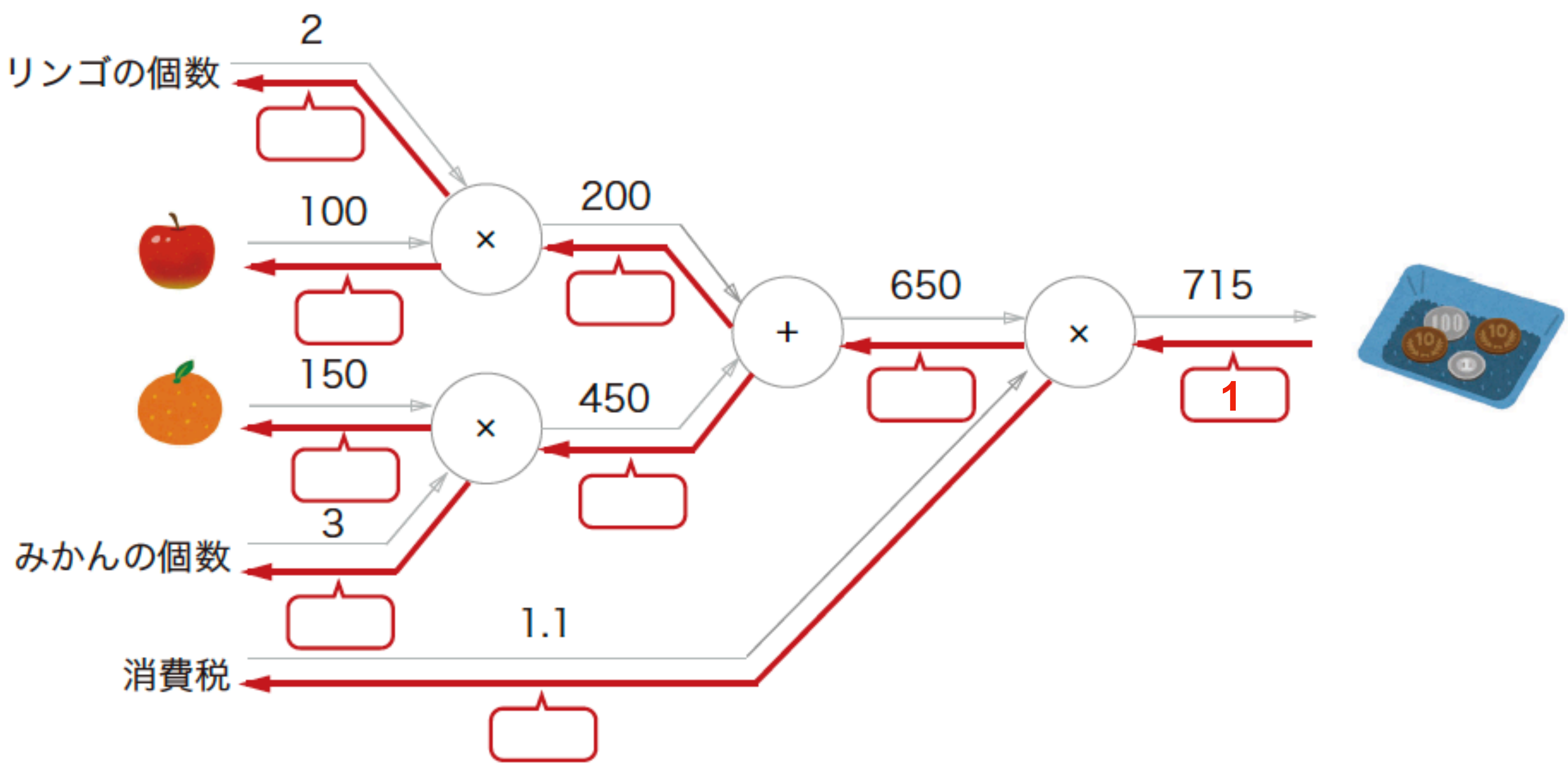
Example: Paid for apple



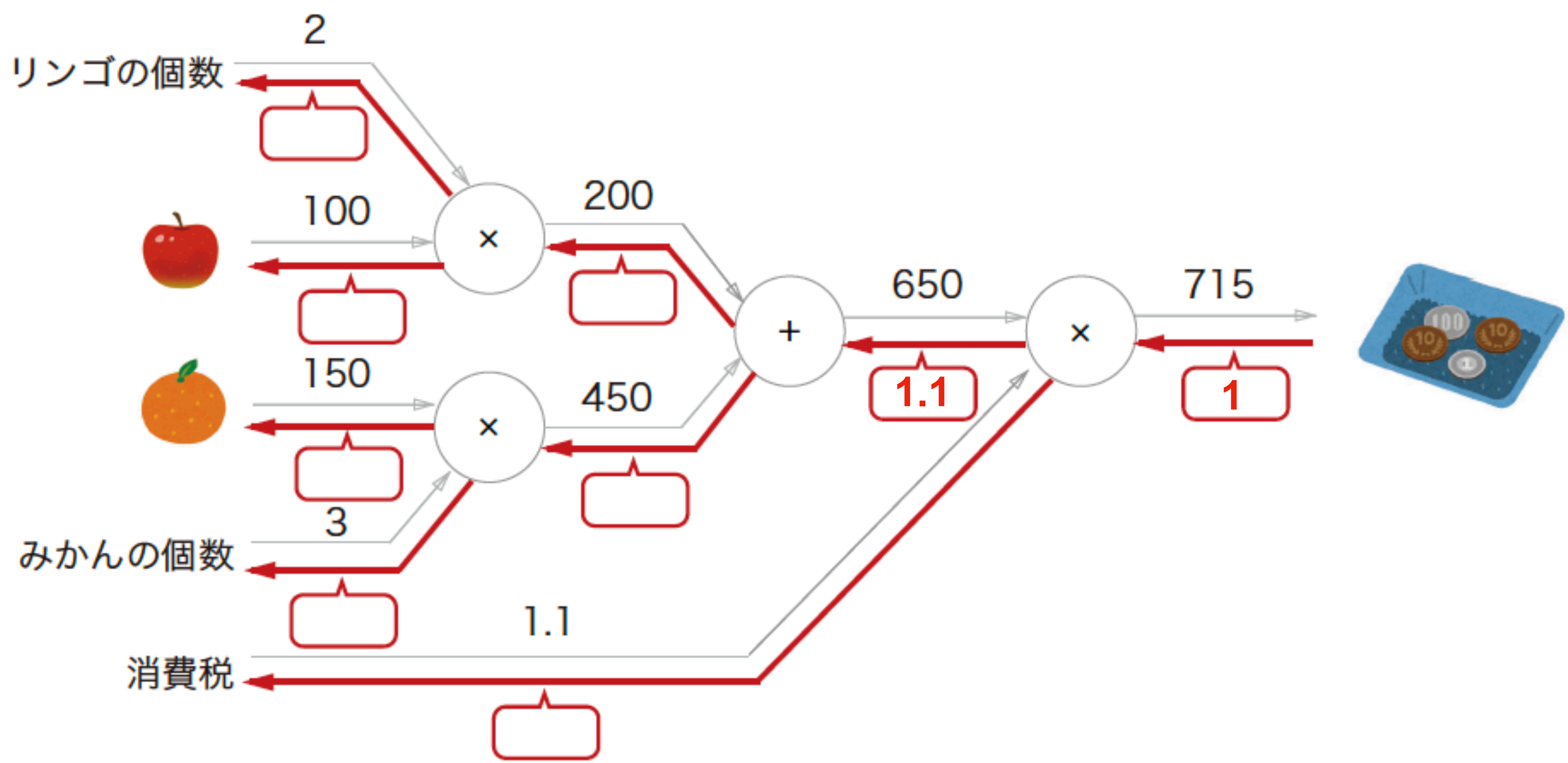
Practice: Apples and Oranges



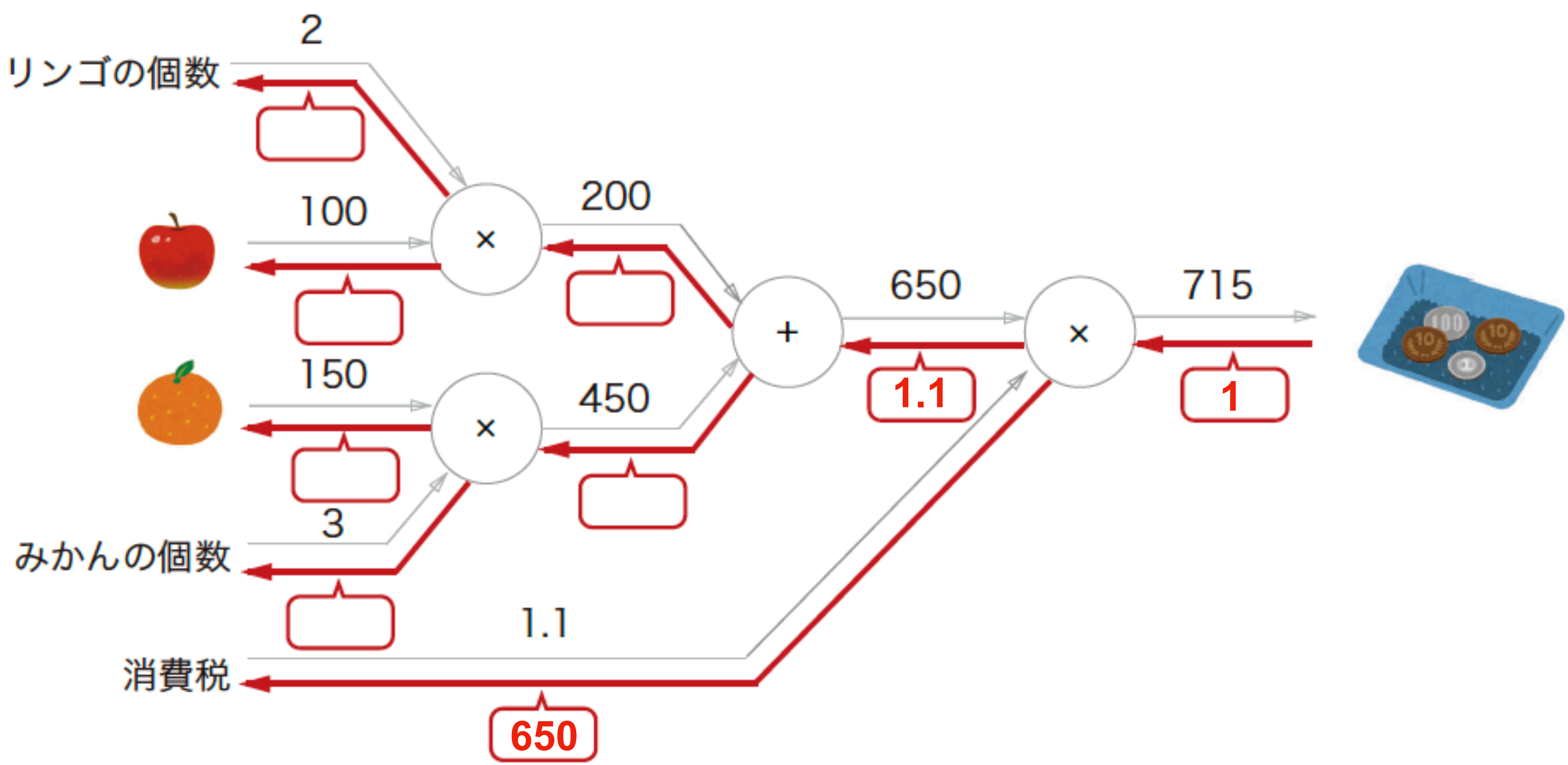
Practice: Apples and Oranges



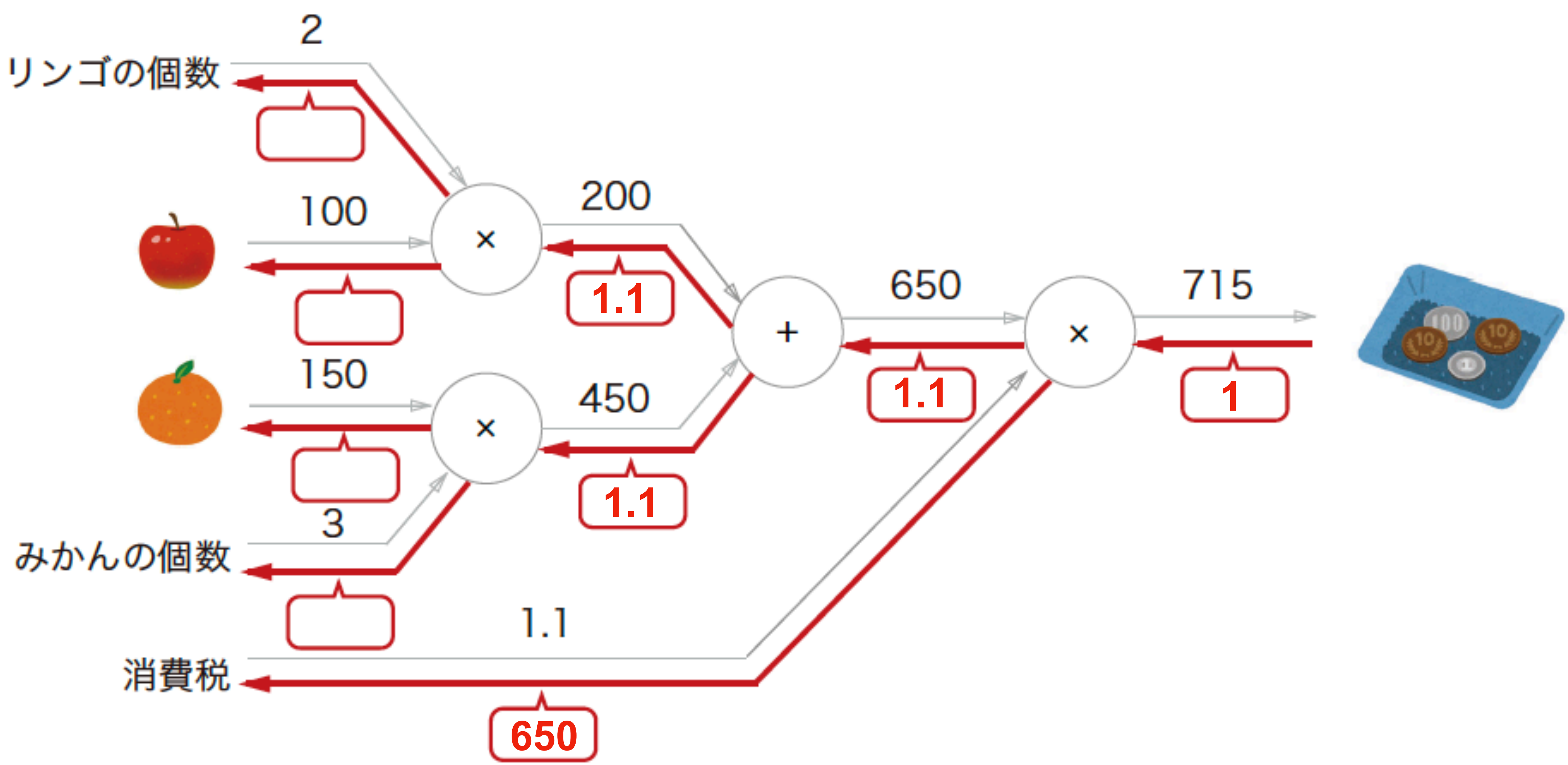
Practice: Apples and Oranges



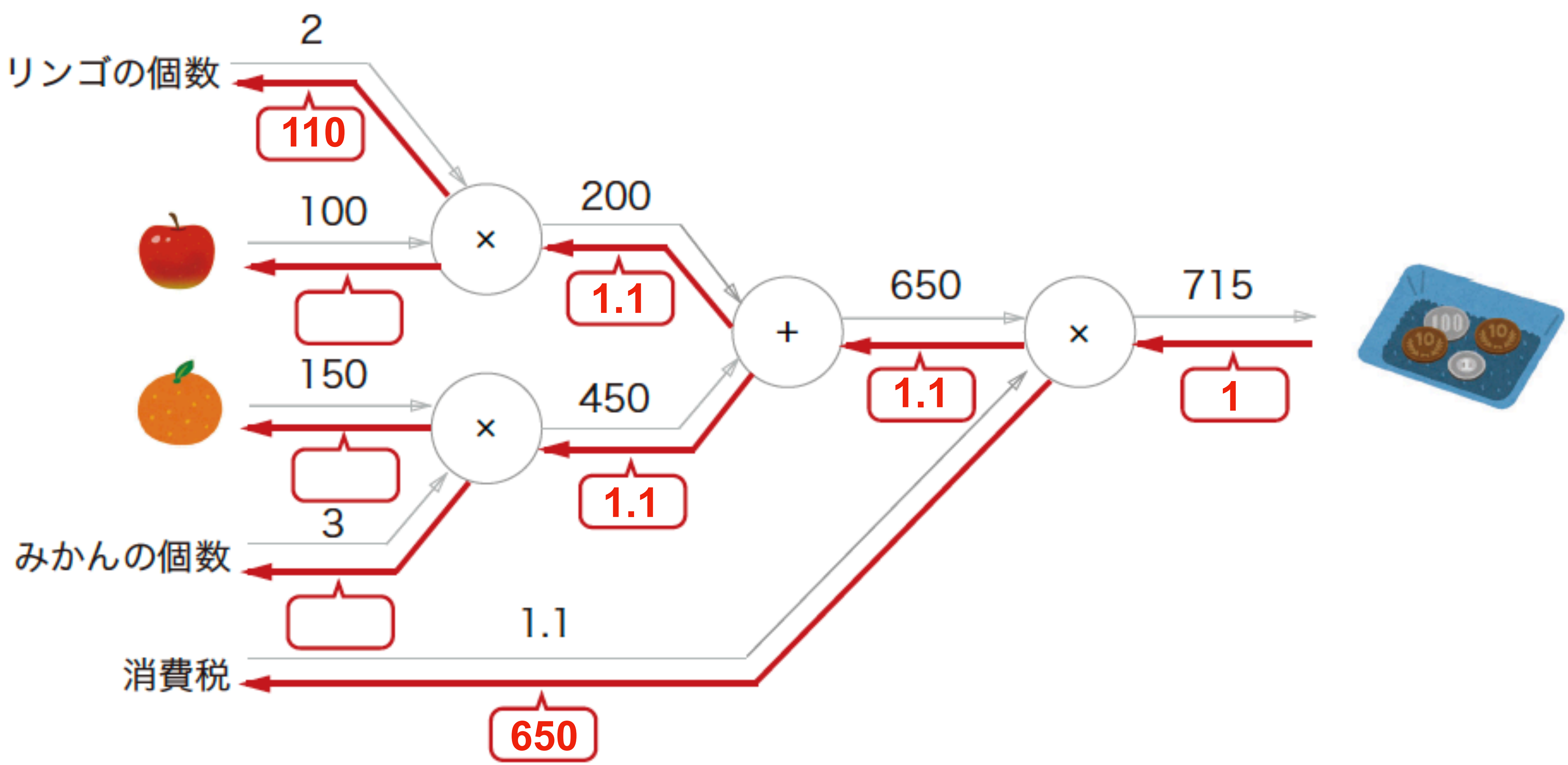
Practice: Apples and Oranges



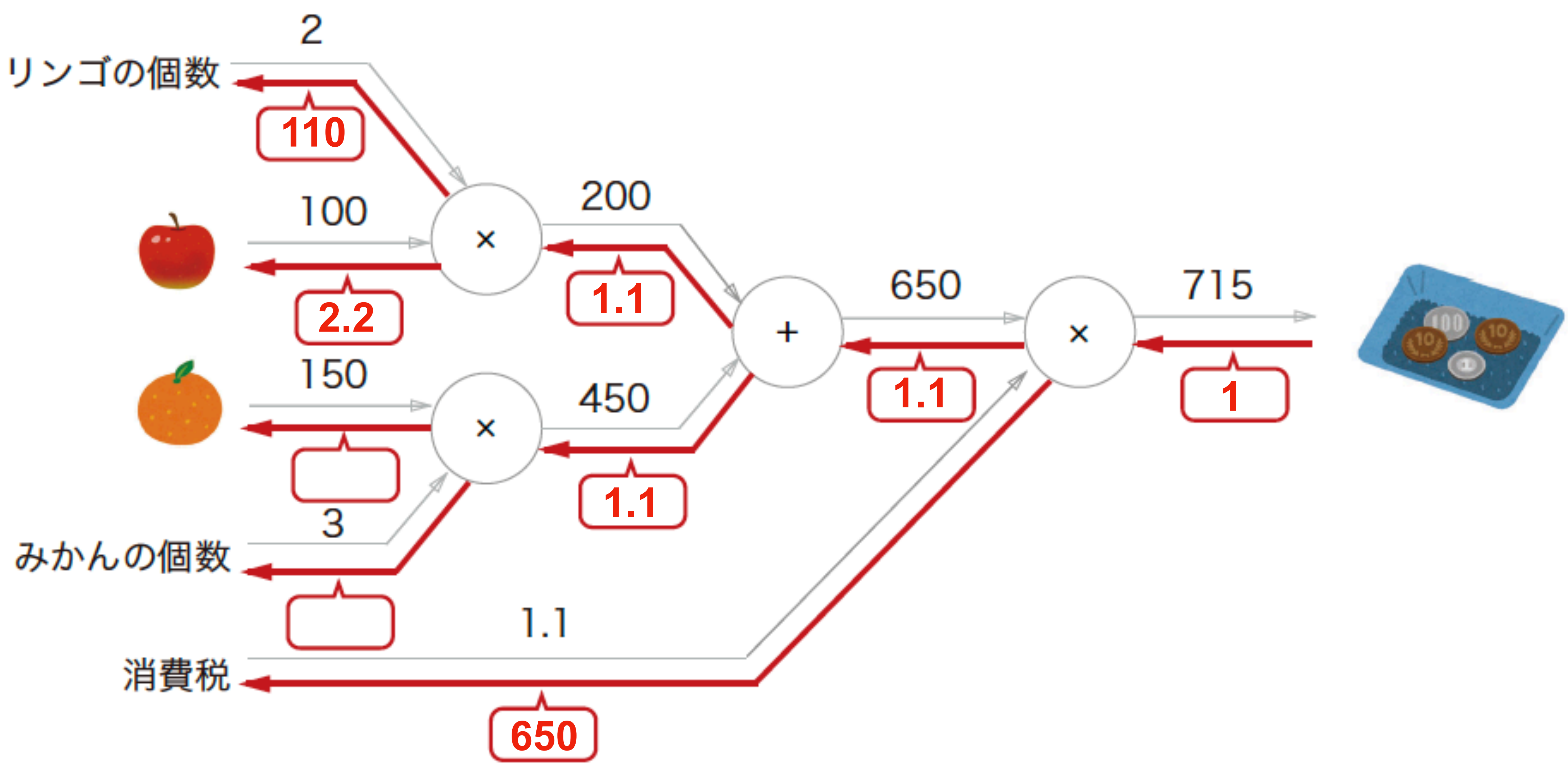
Practice: Apples and Oranges



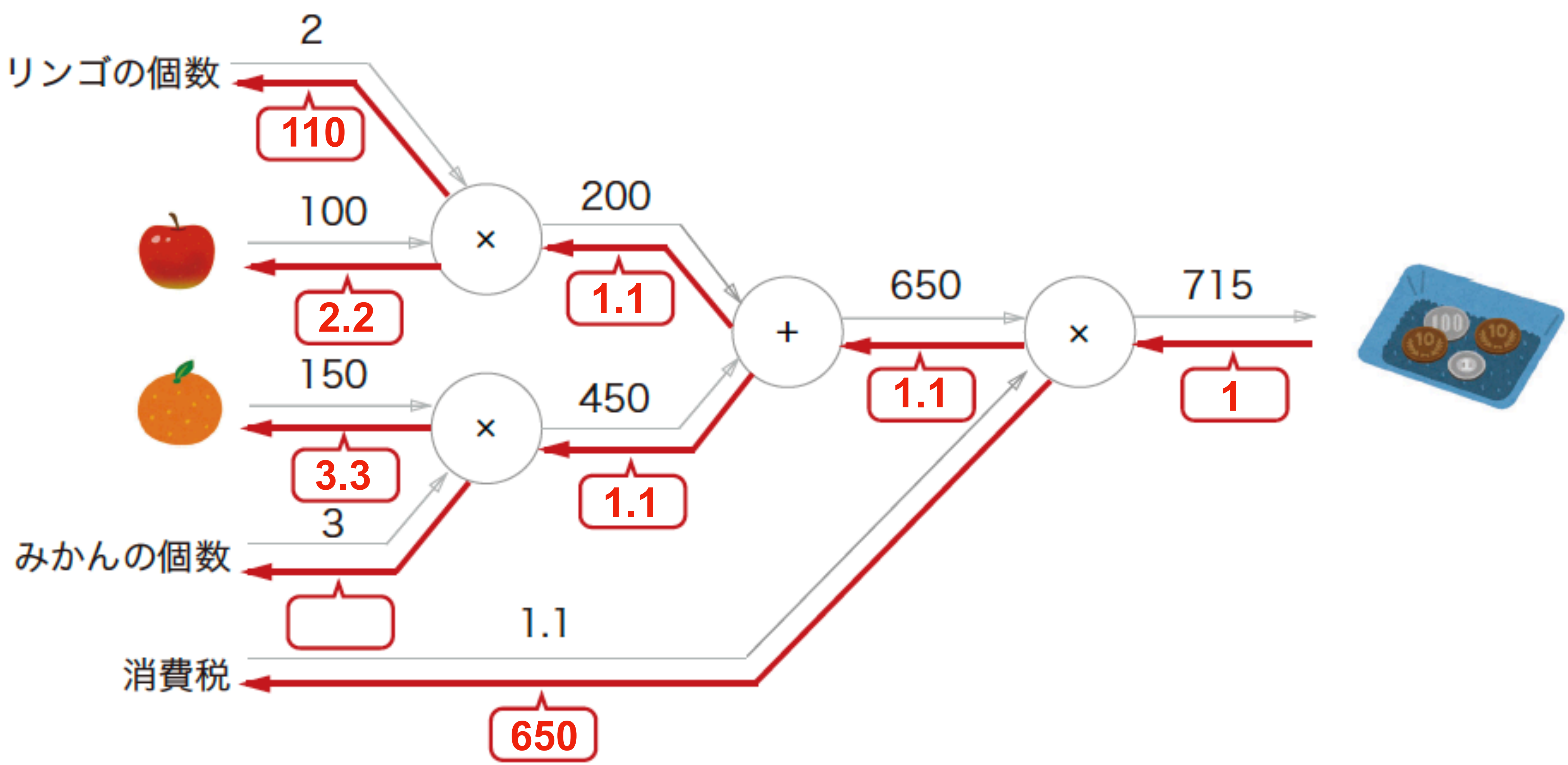
Practice: Apples and Oranges



Practice: Apples and Oranges



Practice: Apples and Oranges



Practice: Apples and Oranges