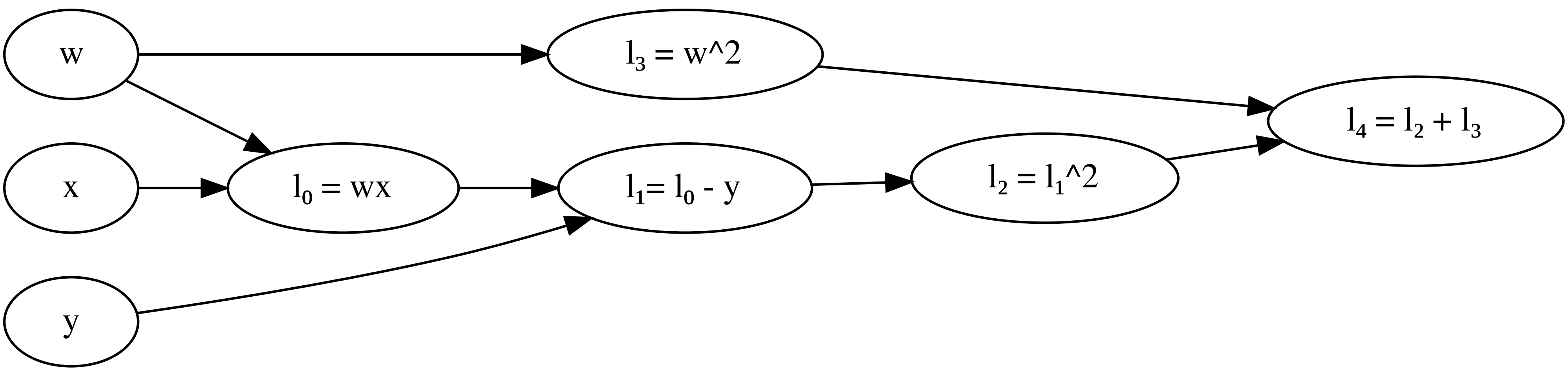
Reverse Mode Automatic Differentiation

Nathan Sprague
James Madison University
CS 445 Machine Learning

Department of Computer Science

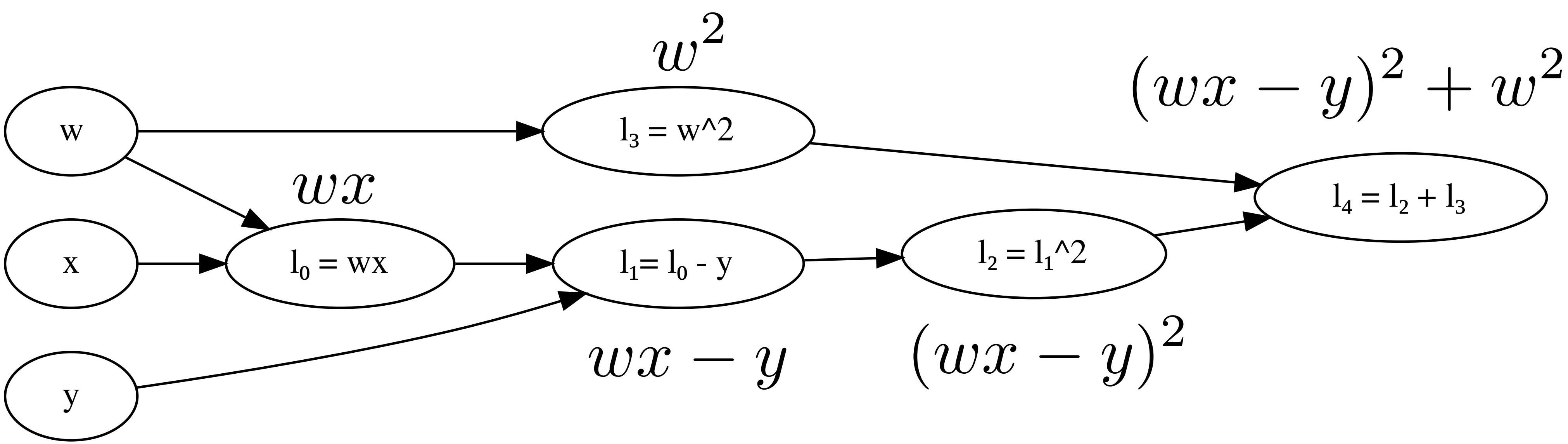
Introduction to Autograd/Reverse-Mode Automatic Differentiation

- Key idea: represent numerical computations using a graph.
- For example: $L(w, x, y) = (wx - y)^2 + w^2$



Introduction to Autograd/Reverse-Mode Automatic Differentiation

- Key idea: represent numerical computations using a graph.
- For example: $L(w, x, y) = (wx - y)^2 + w^2$

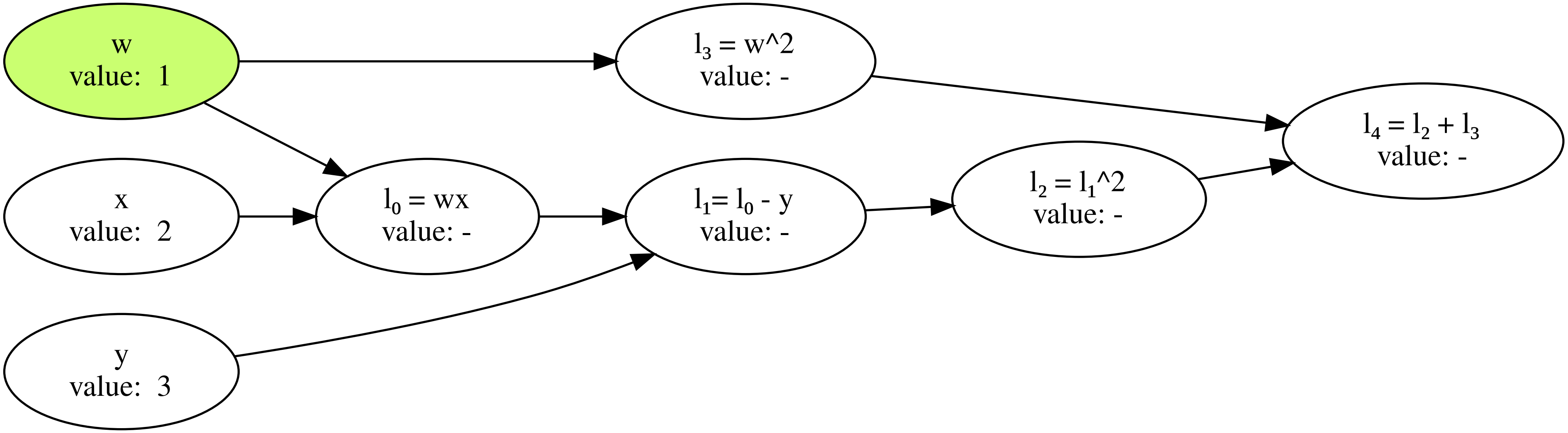


Forward Pass

- Perform a topological sort
- Iterate forward...

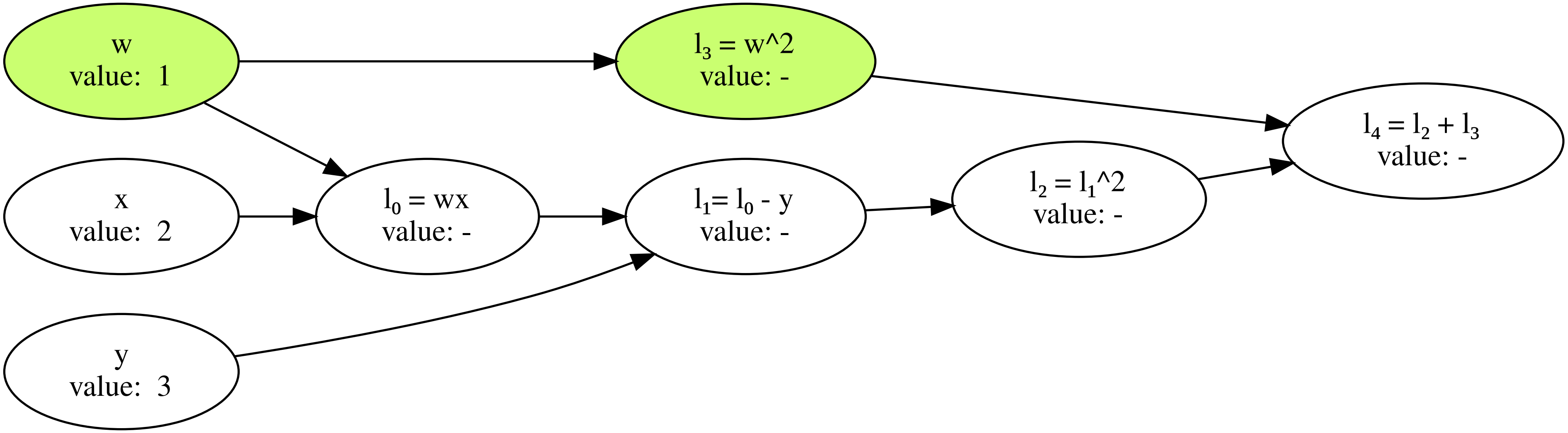
Forward Pass

- Perform a topological sort
- Iterate forward...



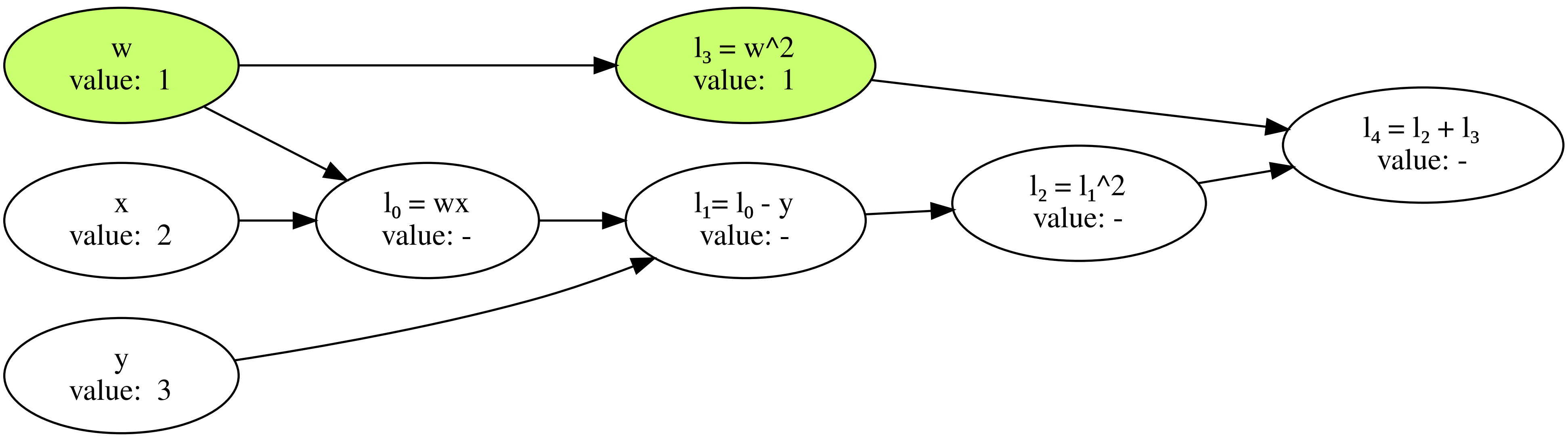
Forward Pass

- Perform a topological sort
- Iterate forward...



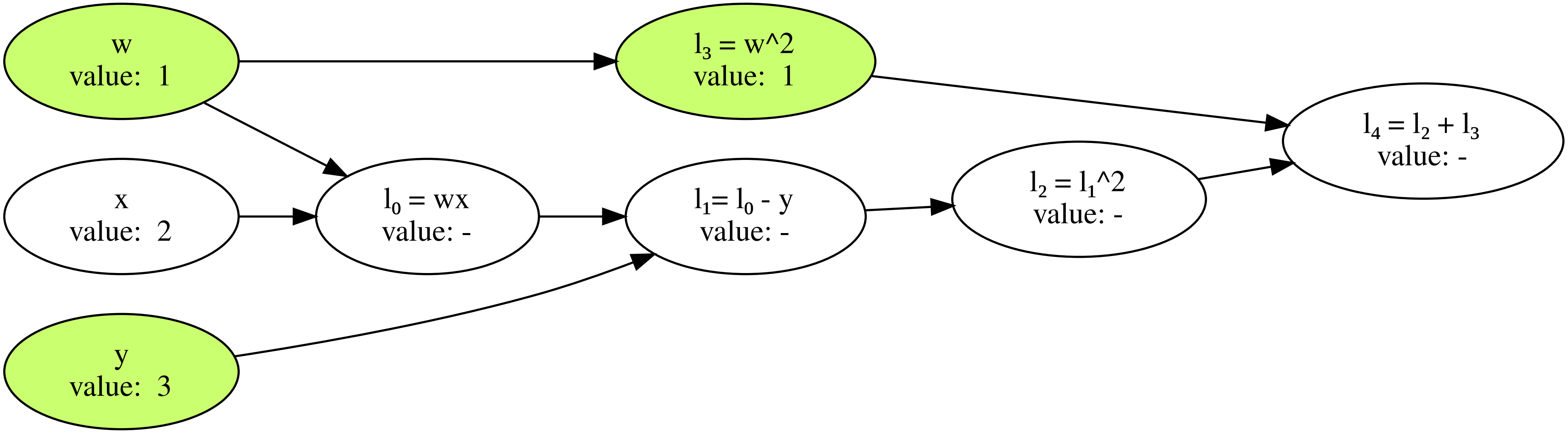
Forward Pass

- Perform a topological sort
- Iterate forward...



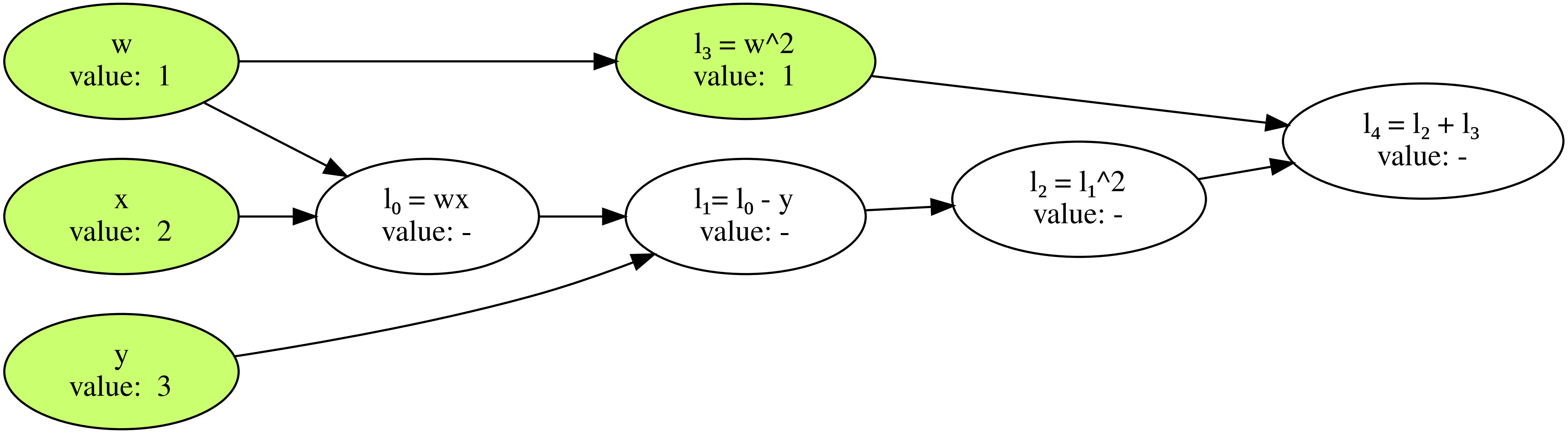
Forward Pass

- Perform a topological sort
- Iterate forward...



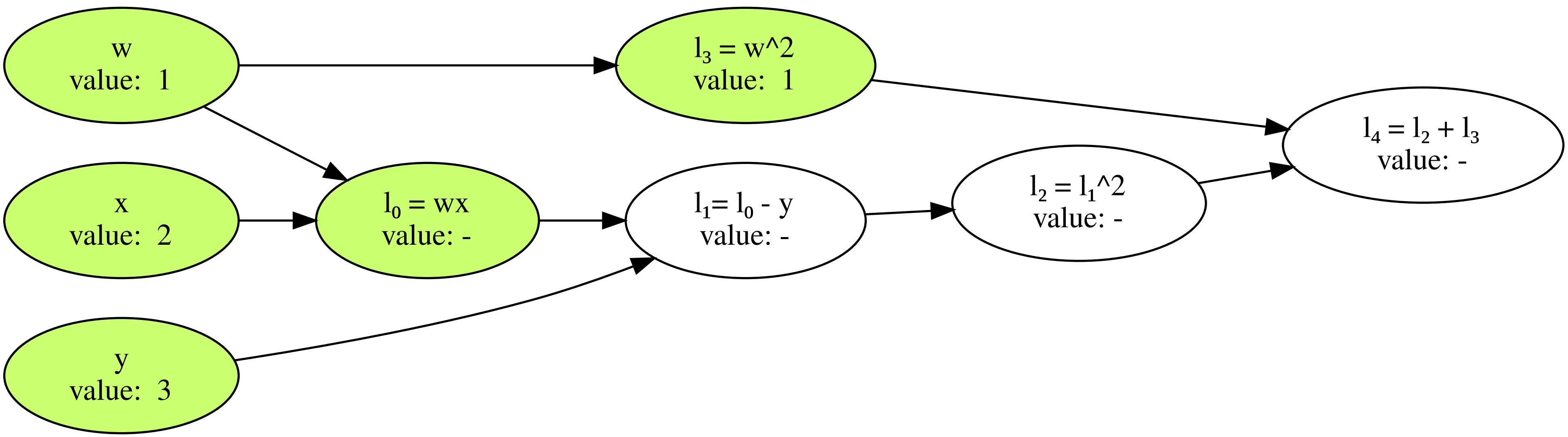
Forward Pass

- Perform a topological sort
- Iterate forward...



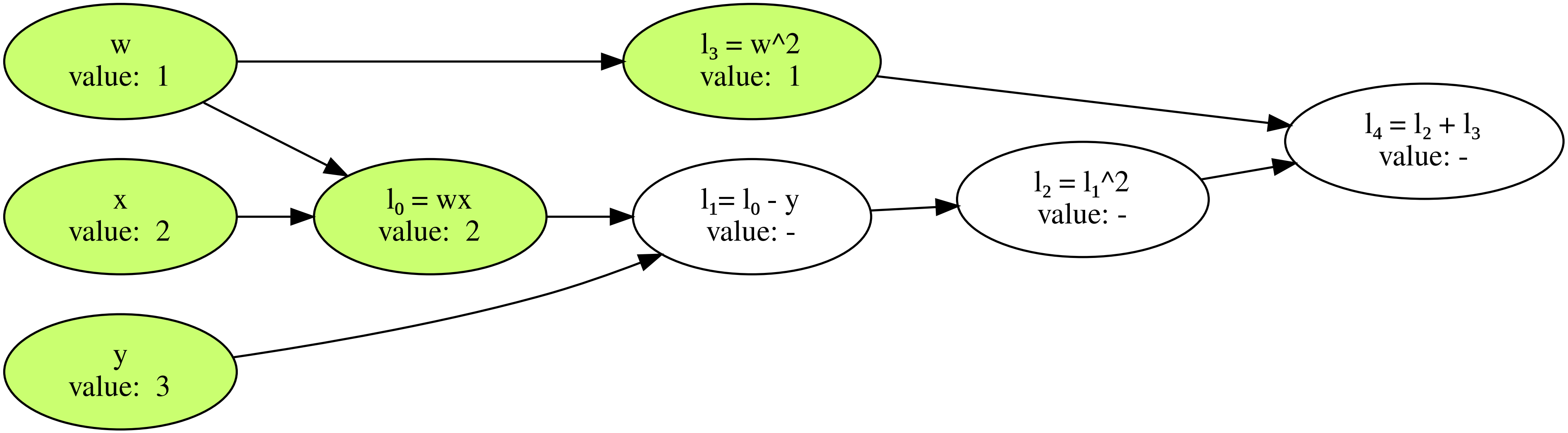
Forward Pass

- Perform a topological sort
- Iterate forward...



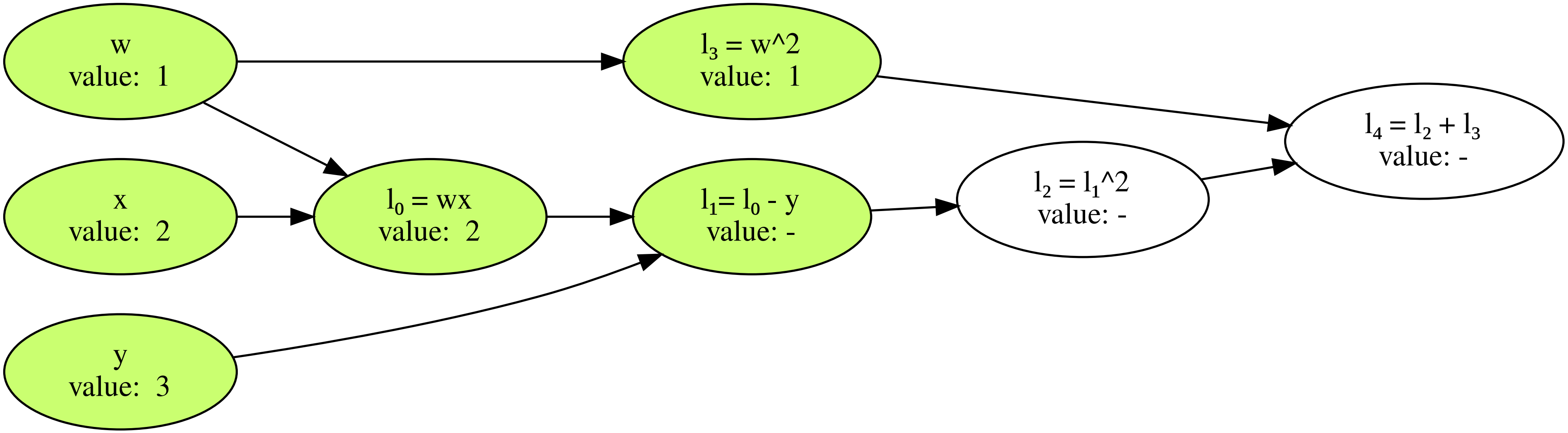
Forward Pass

- Perform a topological sort
- Iterate forward...



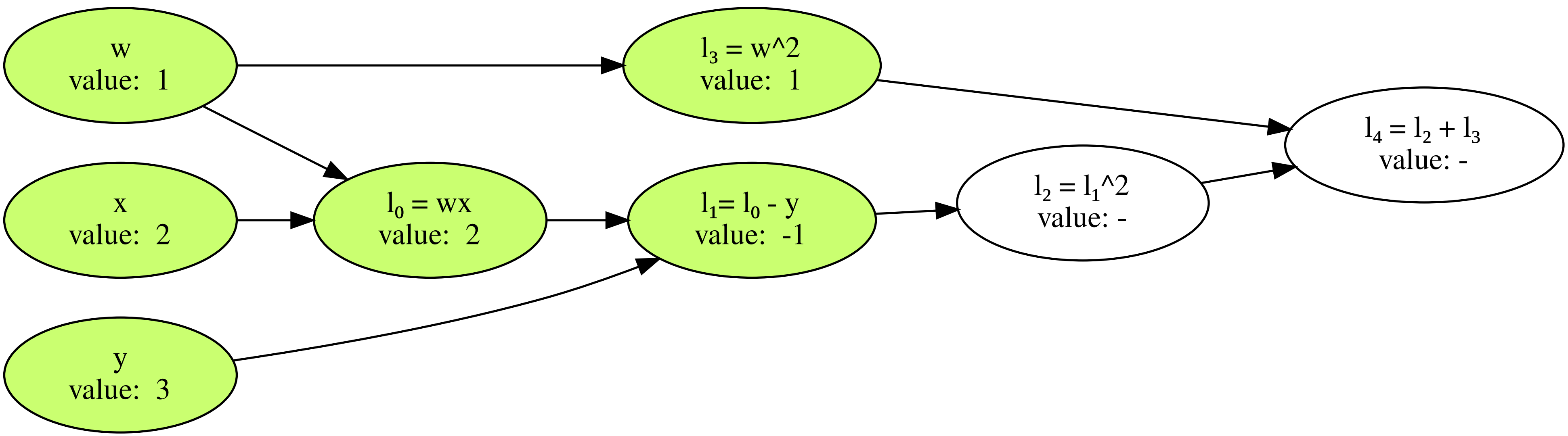
Forward Pass

- Perform a topological sort
- Iterate forward...



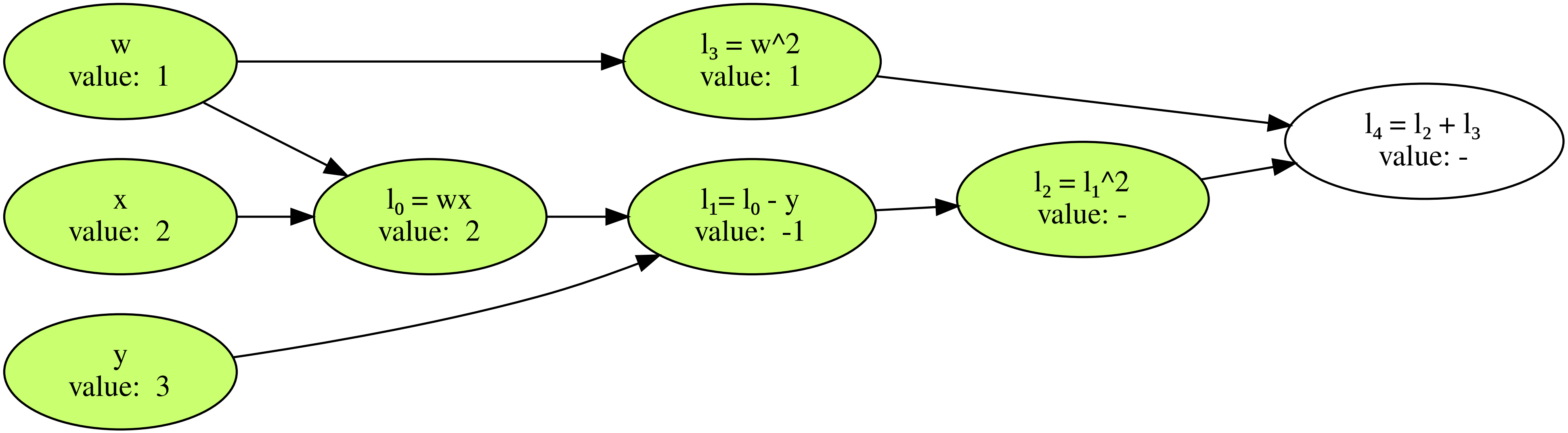
Forward Pass

- Perform a topological sort
- Iterate forward...



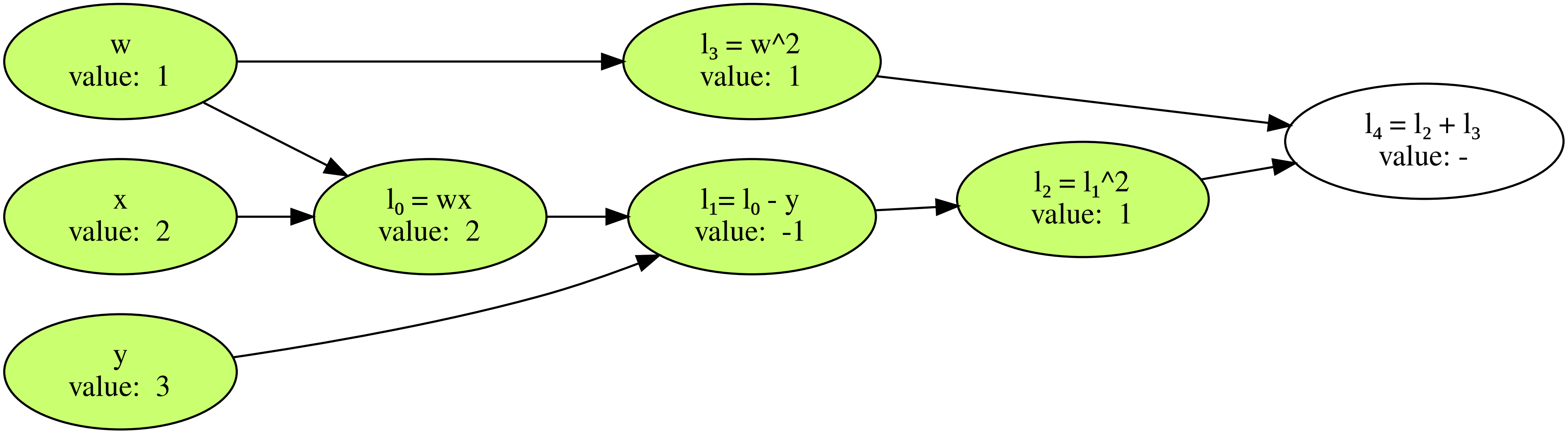
Forward Pass

- Perform a topological sort
- Iterate forward...



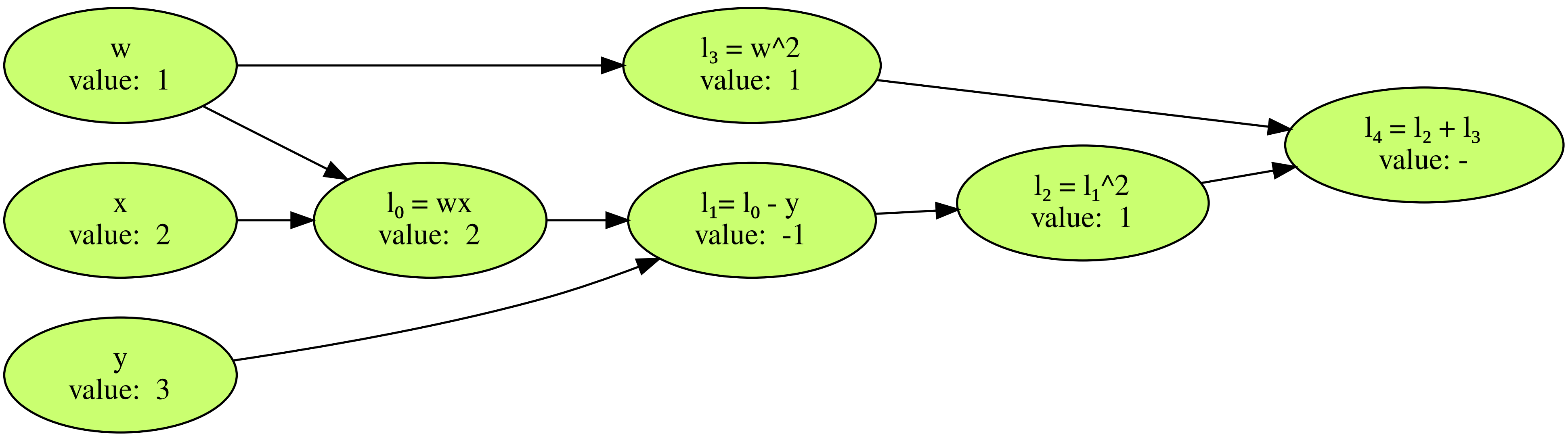
Forward Pass

- Perform a topological sort
- Iterate forward...



Forward Pass

- Perform a topological sort
- Iterate forward...

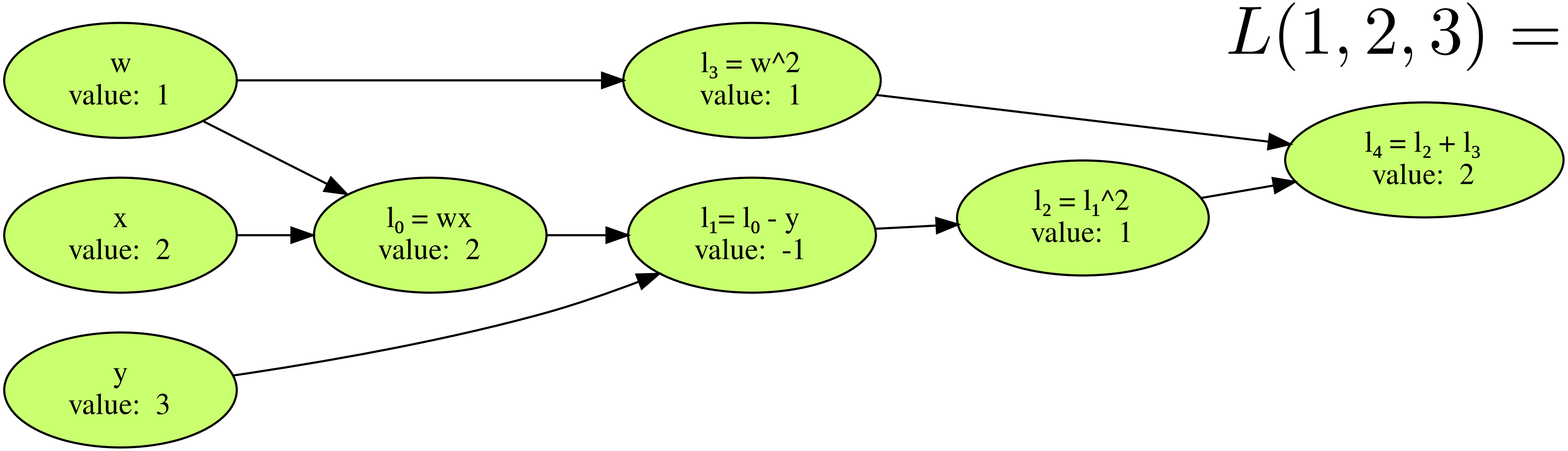


Forward Pass

- Perform a topological sort
- Iterate forward...

$$L(w, x, y) = (wx - y)^2 + w^2$$

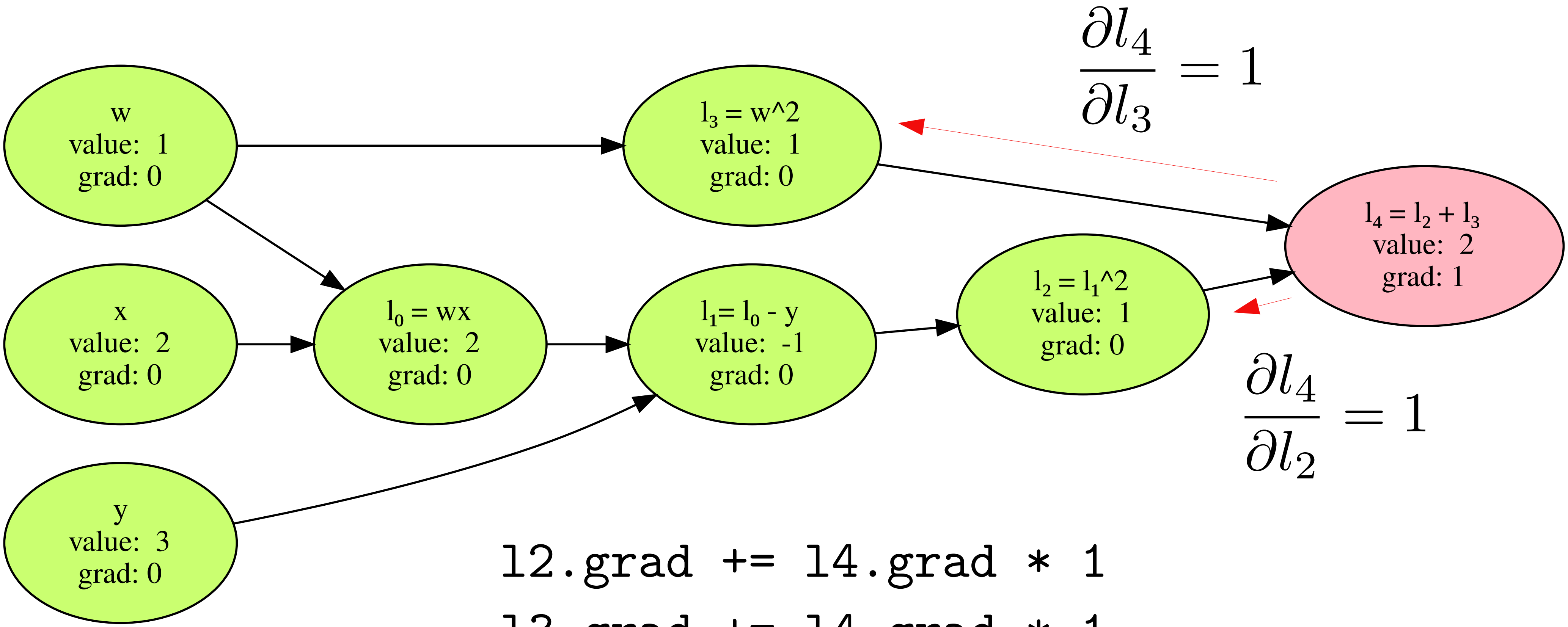
$$L(1, 2, 3) = 2$$



Backward Pass

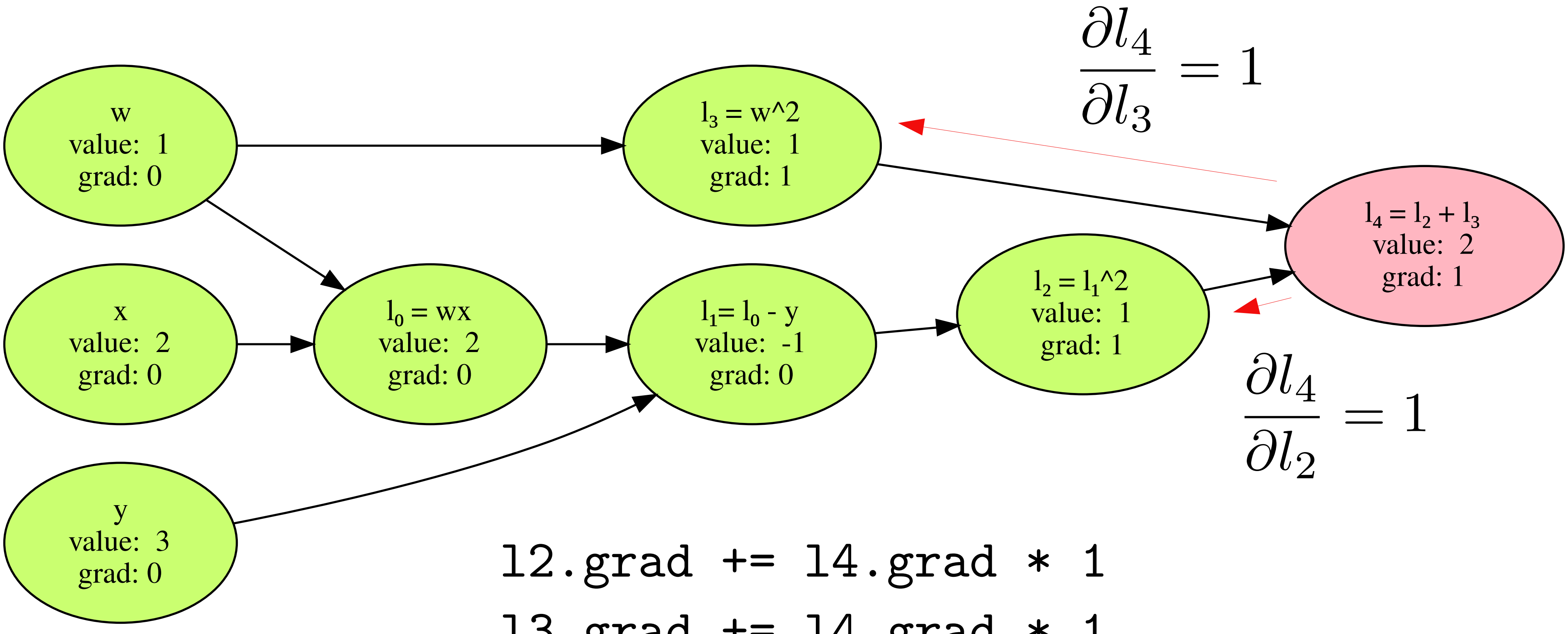
- Perform a reverse topological sort of the ancestors of the node we want to differentiate
- Iterate backwards, accumulating derivative values using the chain rule

Backward Pass



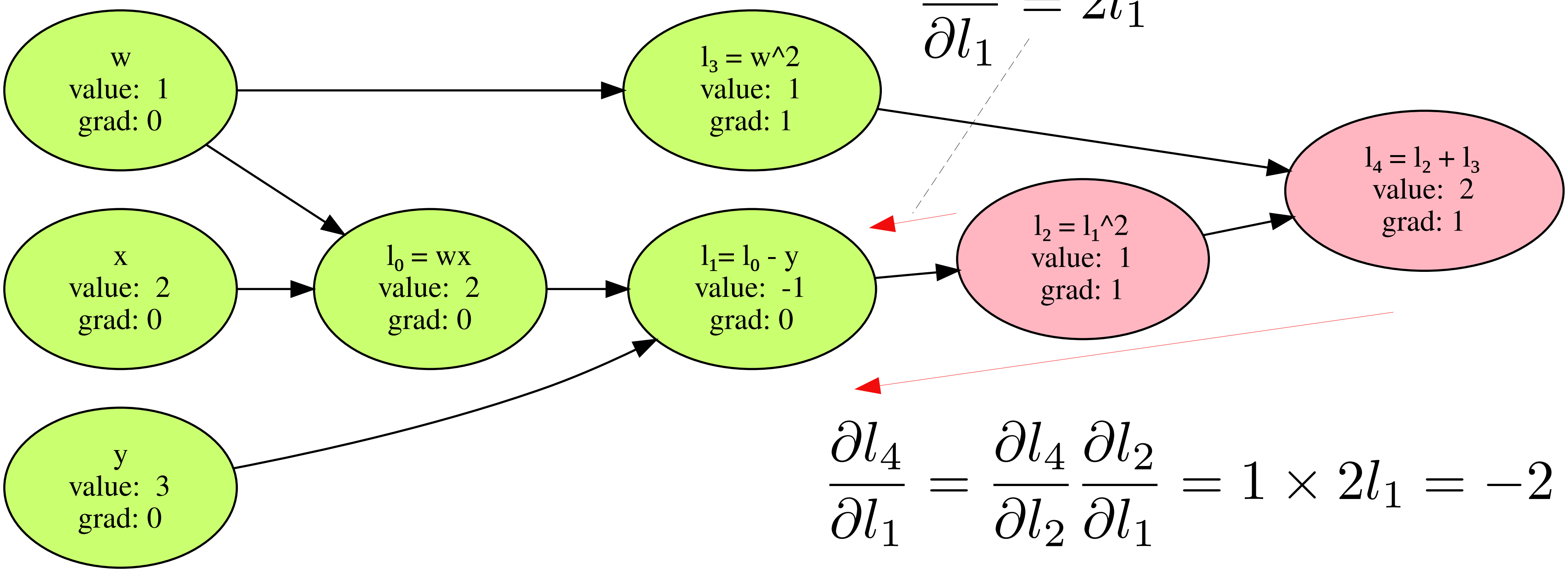

```
l2.grad += l4.grad * 1
l3.grad += l4.grad * 1
```

Backward Pass




```
l2.grad += l4.grad * 1
l3.grad += l4.grad * 1
```

Backward Pass

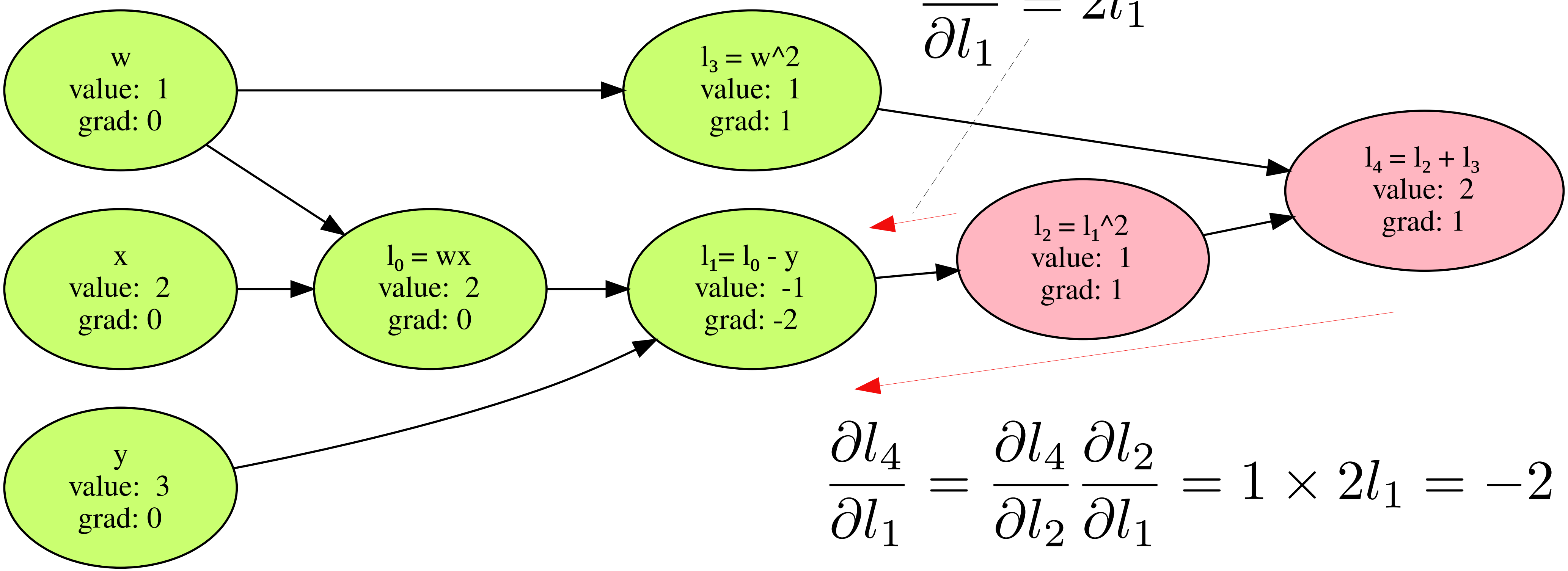


$$\frac{\partial l_2}{\partial l_1} = 2l_1$$

$$\frac{\partial l_4}{\partial l_1} = \frac{\partial l_4}{\partial l_2}\frac{\partial l_2}{\partial l_1} = 1 \times 2l_1 = -2$$

```
l1.grad += l2.grad * 2 * l1.value
```

Backward Pass

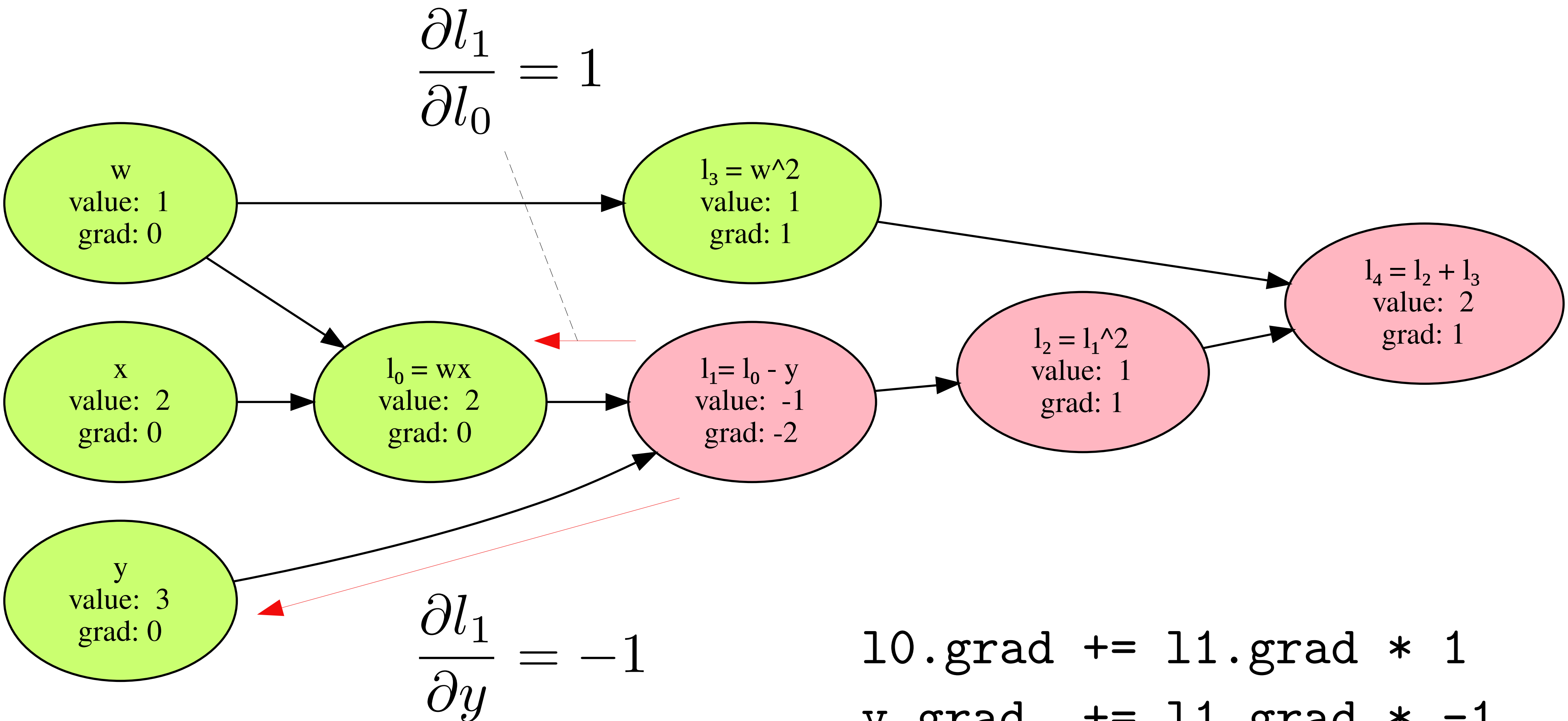


$$\frac{\partial l_2}{\partial l_1} = 2l_1$$

$$\frac{\partial l_4}{\partial l_1} = \frac{\partial l_4}{\partial l_2}\frac{\partial l_2}{\partial l_1} = 1 \times 2l_1 = -2$$

```
l1.grad += l2.grad * 2 * l1.value
```

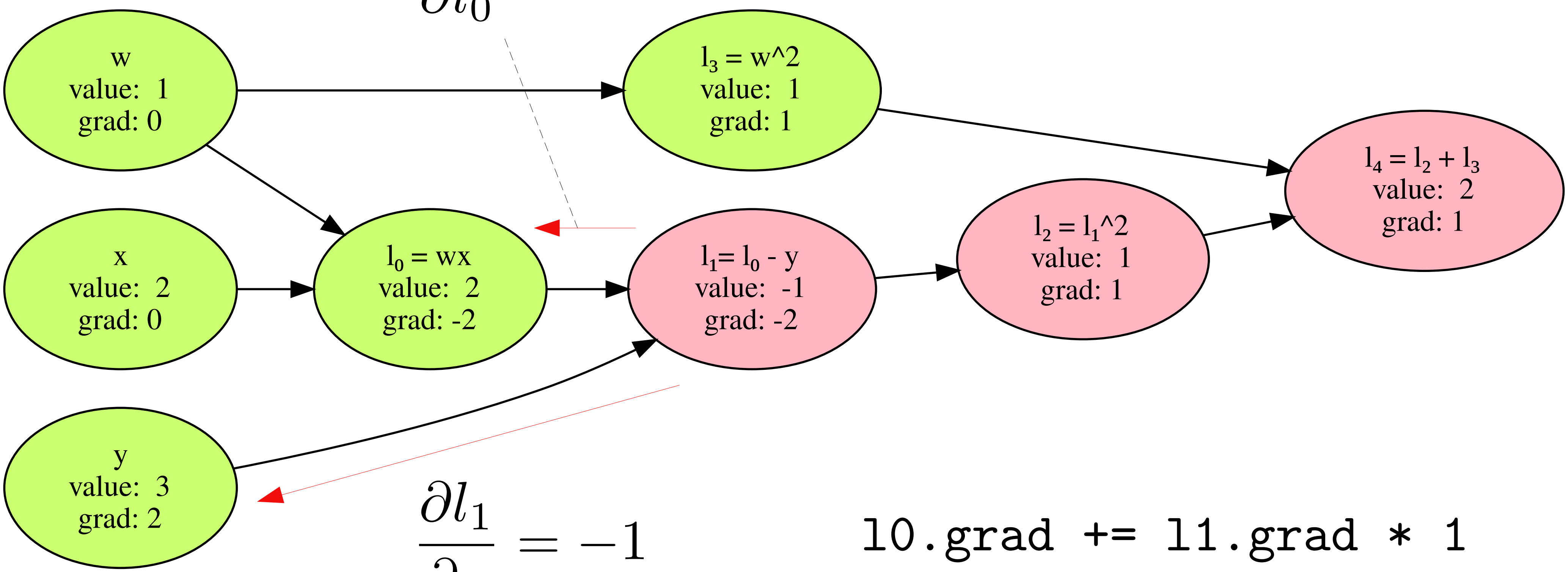
Backward Pass




```
l0.grad += l1.grad * 1
y.grad  += l1.grad * -1
```

Backward Pass

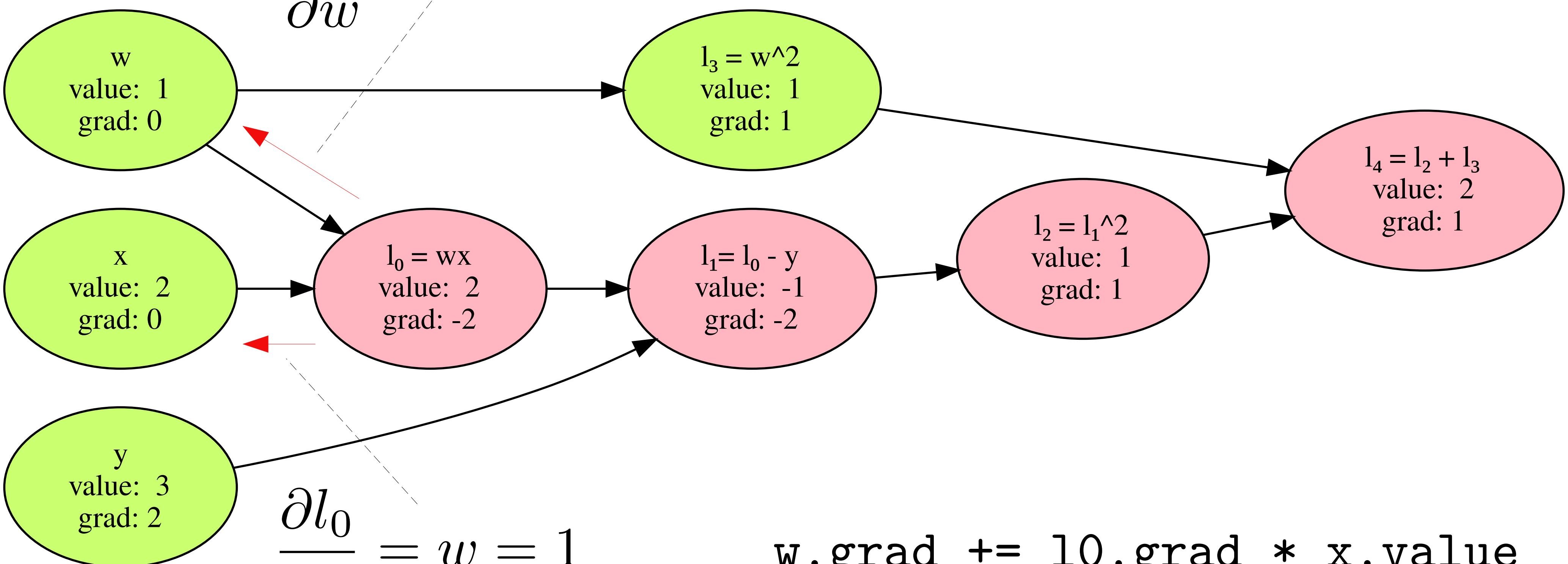
$$\frac{\partial l_1}{\partial l_0} = 1$$



$$\frac{\partial l_1}{\partial y} = -1$$

```
l0.grad += l1.grad * 1
y.grad  += l1.grad * -1
```

Backward Pass



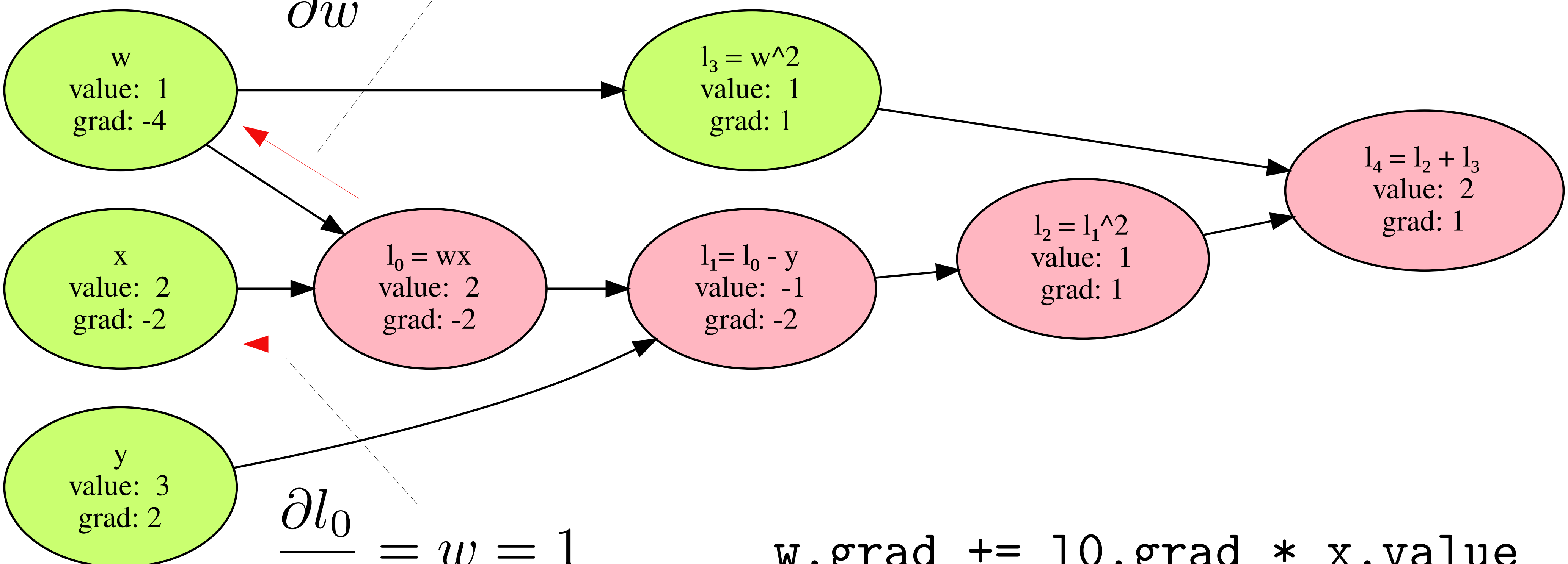
$$\frac{\partial l_0}{\partial w} = x = 2$$

$$\frac{\partial l_0}{\partial x} = w = 1$$

```
w.grad += l0.grad * x.value
x.grad += l0.grad * w.value
```

Backward Pass

$$\frac{\partial l_0}{\partial w} = x = 2$$

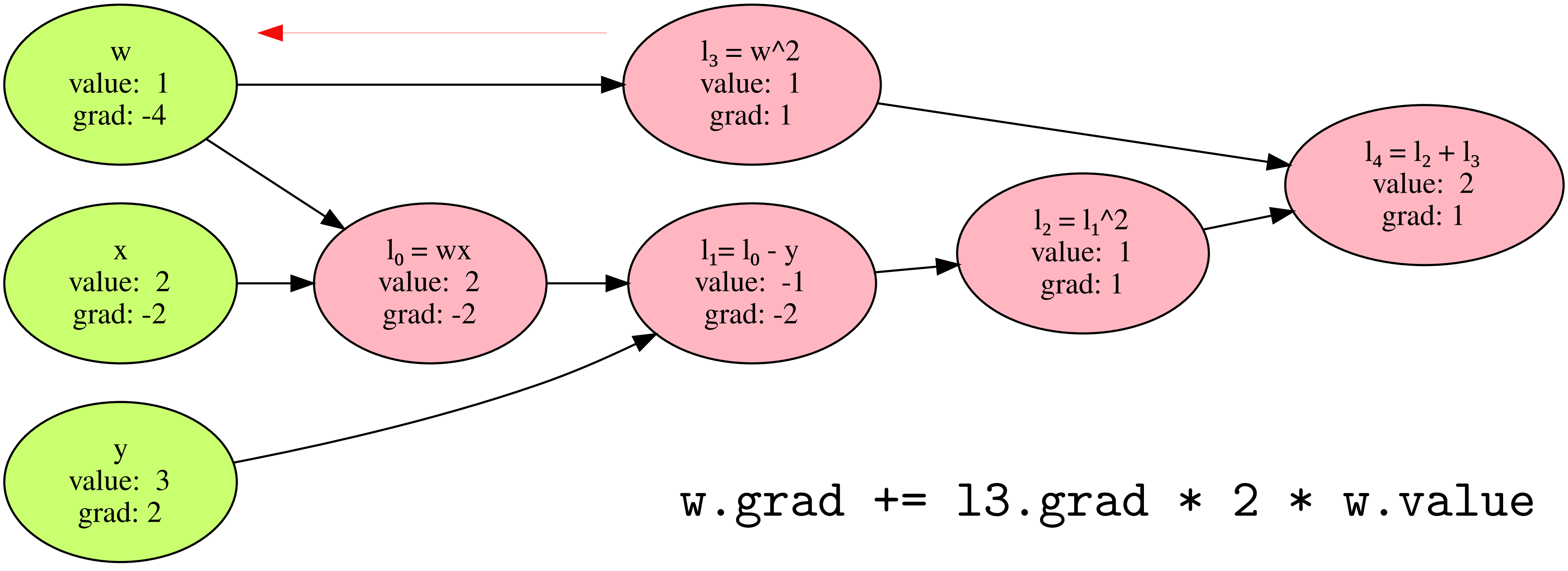


$$\frac{\partial l_0}{\partial x} = w = 1$$

```
w.grad += l0.grad * x.value
x.grad += l0.grad * w.value
```

Backward Pass

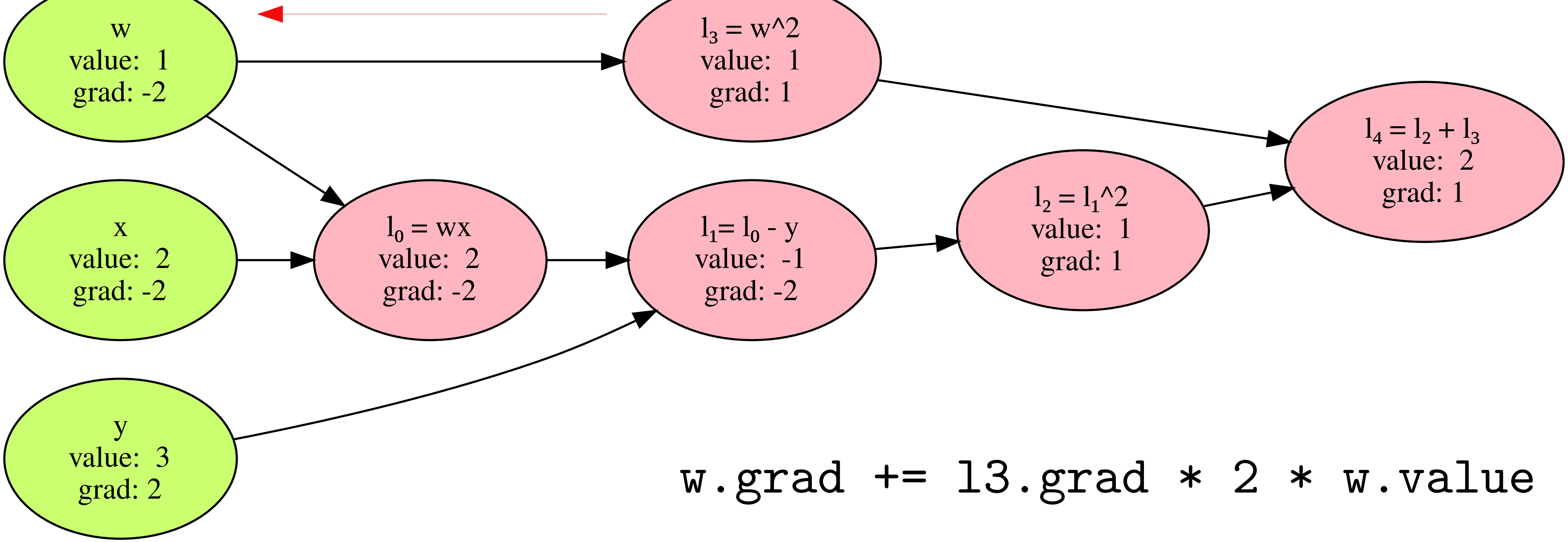
$$\frac{\partial l_3}{\partial w} = 2w = 2$$




```
w.grad += l3.grad * 2 * w.value
```

Backward Pass

$$\frac{\partial l_3}{\partial w} = 2w = 2$$



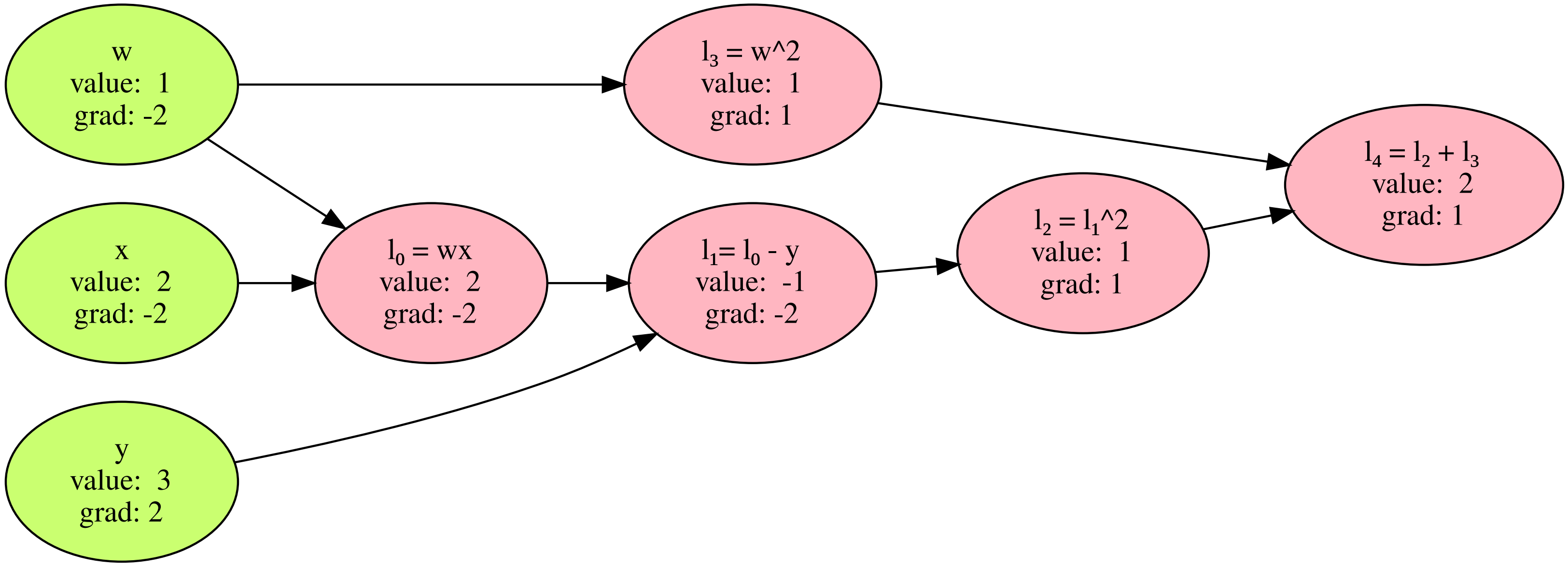

```
w.grad += l3.grad * 2 * w.value
```

Backward Pass

$$L(w, x, y) = (wx - y)^2 + w^2$$

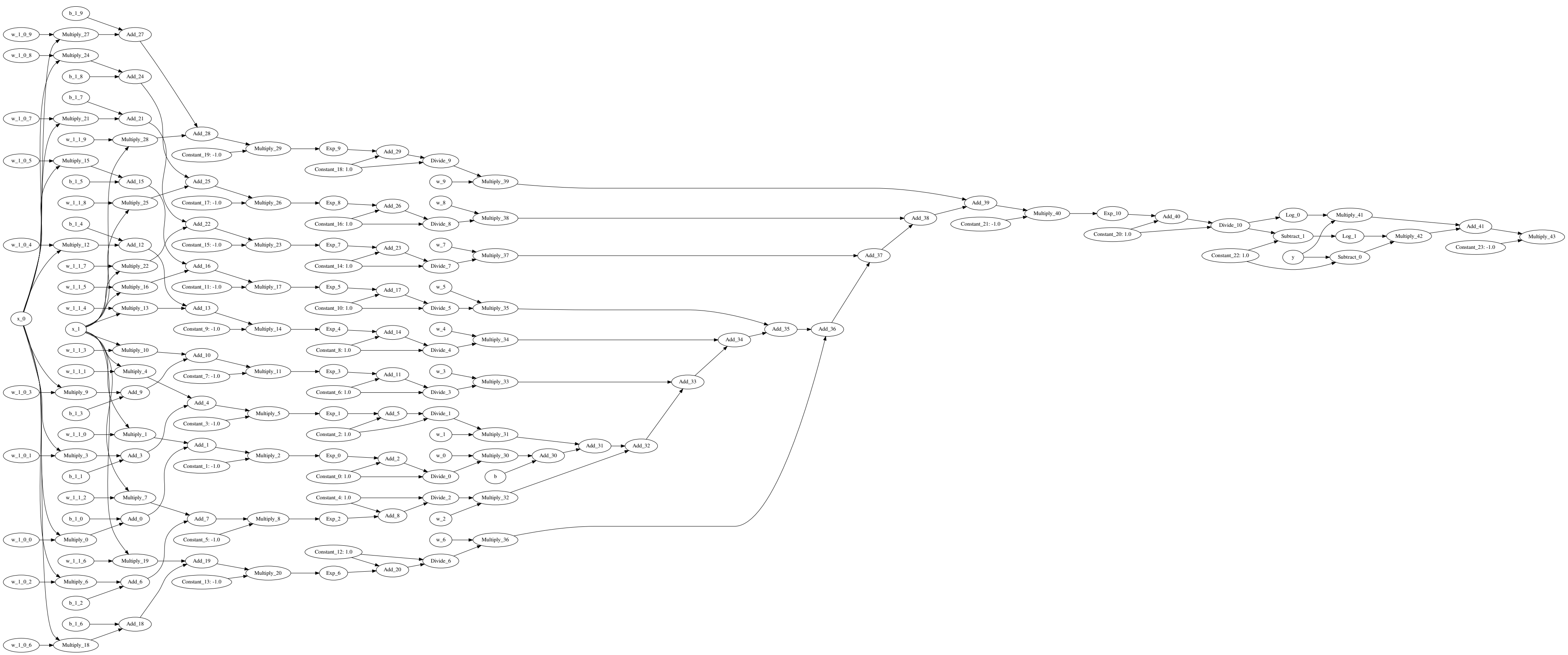
$$\frac{\partial L(w, x, y)}{\partial w} = 2x(wx - y) + 2w$$

$$\frac{\partial L(1, 2, 3)}{\partial w} = -2$$



Classifier Graphs

- Three-layer network with two input units and ten hidden units:



Autodiff in Code

- Two basic approaches to designing and autodiff library:
  - Define-and-run / static computational graph
    - We use library code to build a graph structure, then perform computations using that graph.
      - Theano, TensorFlow 1.0, Caffe
  - Define-by-run / dynamic computational graph
    - We instrument normal code in such a way that the graph is built implicitly during execution.
      - PyTorch, Tensorflow 2.0

Resources

Online tutorials (ordered from less to more detail)

- **Automatic Differentiation, Explained**
  https://towardsdatascience.com/automatic-differentiation-explained-b4ba8e60c2ad
- **Step by Step Example of Reverse Mode Automatic Differentiation**
  https://stats.stackexchange.com/a/235758
- **Reverse-mode automatic differentiation: a tutorial**
  https://rufflewind.com/2016-12-30/reverse-mode-automatic-differentiation

Video tutorials

- **Derivatives with Computation Graphs** (Andrew Ng) https://youtu.be/nJyUyKN-XBQ
- **Gradient and Auto Differentiation** (Alex Smola and Mu Li) https://youtu.be/RP0JScZG6gA

Pedagogical Autograd Implementation

- **Autodidact: a pedagogical implementation of Autograd** https://github.com/mattjj/autodidact

Survey Paper

- Baydin, A.G., Pearlmutter, B.A., Radul, A.A. and Siskind, J.M., 2018. **Automatic differentiation in machine learning: a survey**. Journal of machine learning research, 18.