Automatic Differentiation (2)

Atılım Güneş Baydin
gunes@robots.ox.ac.uk

Summary of last lecture

- Derivatives in machine learning
- Review of essential concepts
  - derivative, partial derivative, total derivative, gradient, Jacobian, matrix calculus, etc.
- How do we compute derivatives
  - Manual, symbolic, numerical
- Automatic differentiation
- Computational graphs and propagation

Today

- The reverse mode (backprop) computational graph
    - What gets propagated?
- Implementation
    - Where does the graph come from?
    - Strategies and performance tips
- Advanced concepts
    - Nesting, higher-order derivatives
    - Reparameterization
    - Checkpointing

Reverse mode computational graph

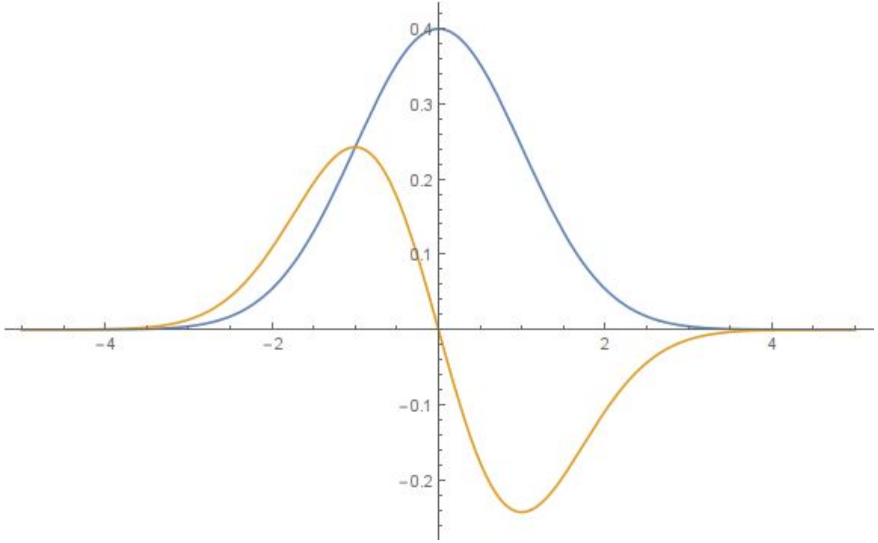
Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Normal PDF

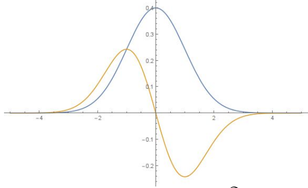
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\frac{\partial f}{\partial x} = \frac{(\mu - x)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3} \qquad \frac{\partial f}{\partial \mu} = \frac{(x - \mu)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3} \qquad \frac{\partial f}{\partial \sigma} = -\frac{(\sigma - x + \mu)(\sigma + x - \mu)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^4}$$

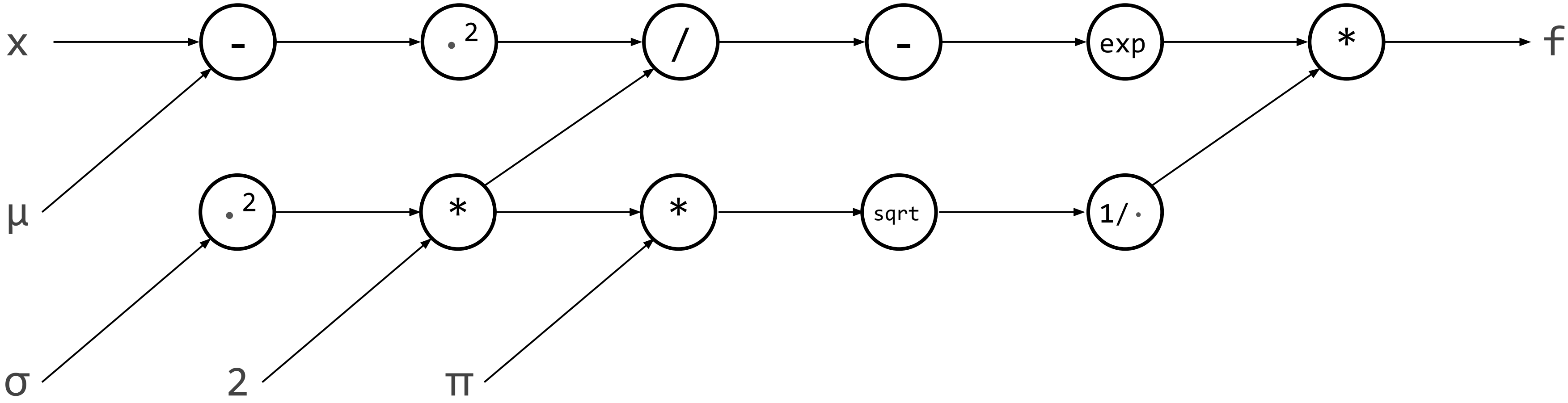


Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

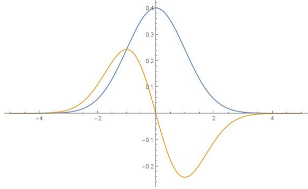


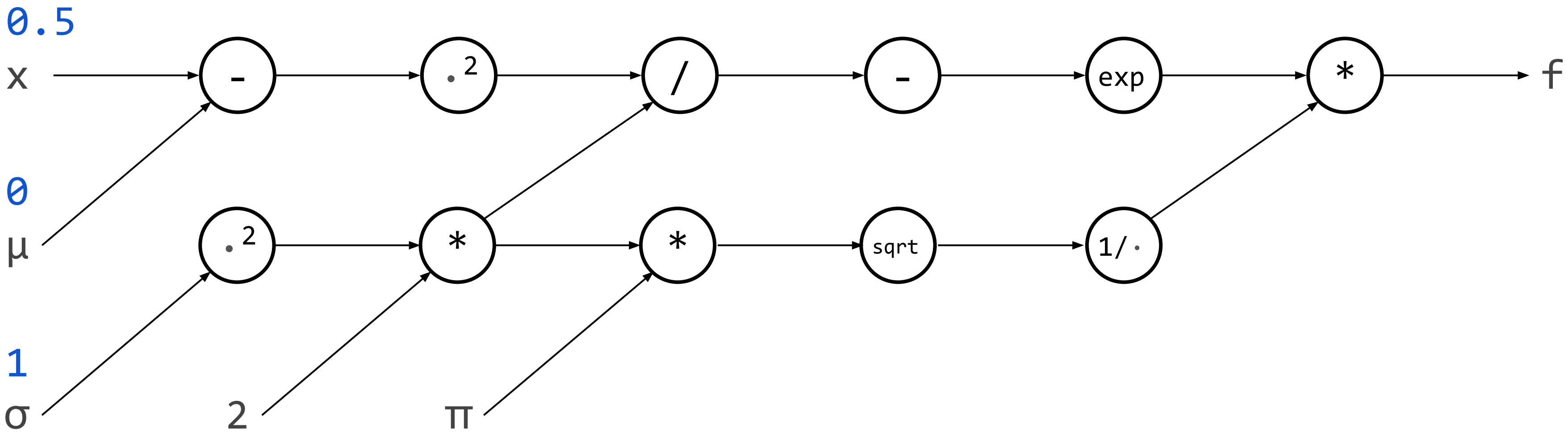
$$\frac{\partial f}{\partial x} = \frac{(\mu - x) e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3} \qquad \frac{\partial f}{\partial \mu} = \frac{(x - \mu) e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3} \qquad \frac{\partial f}{\partial \sigma} = -\frac{(\sigma - x + \mu)(\sigma + x - \mu) e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^4}$$



Normal PDF

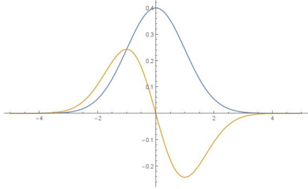
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

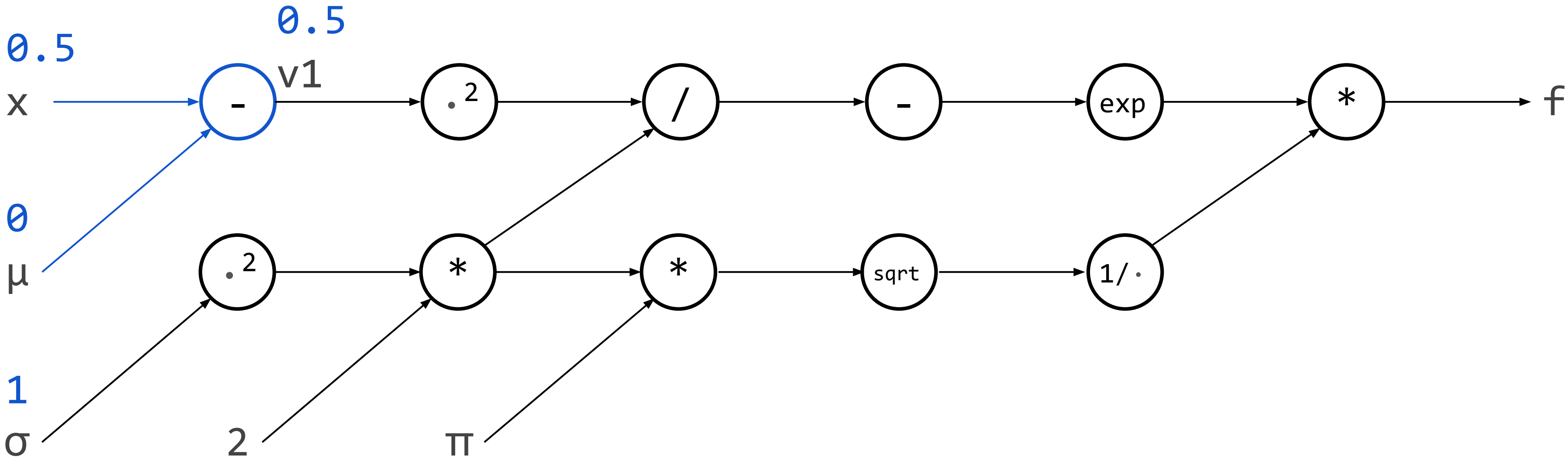




Normal PDF

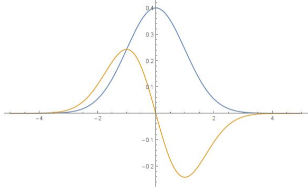
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

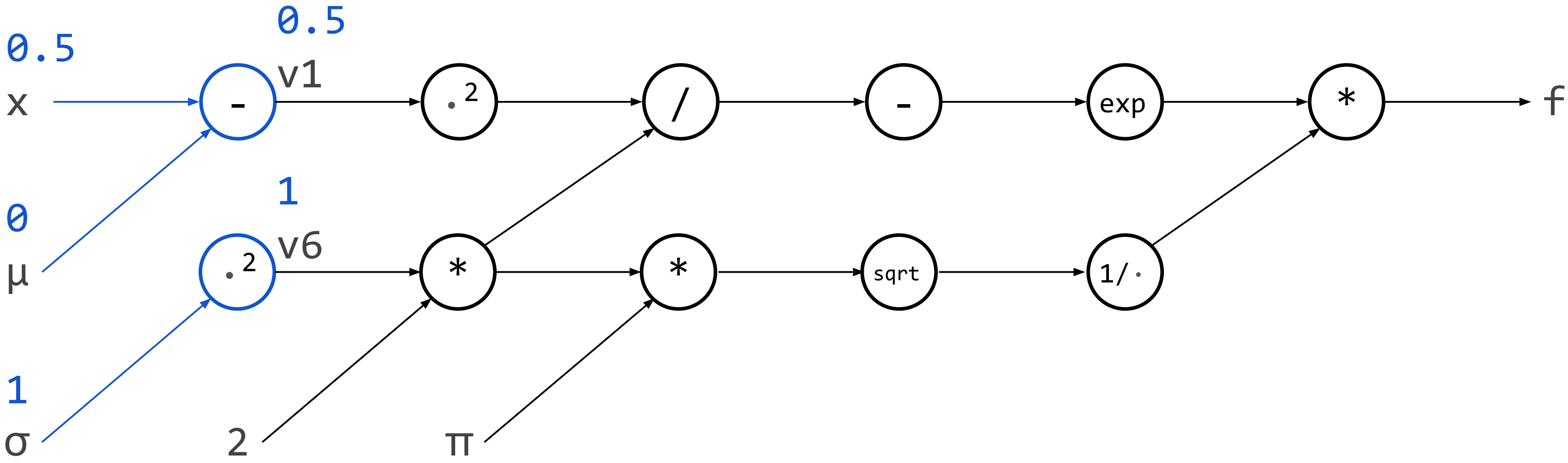




Normal PDF

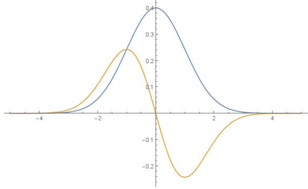
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

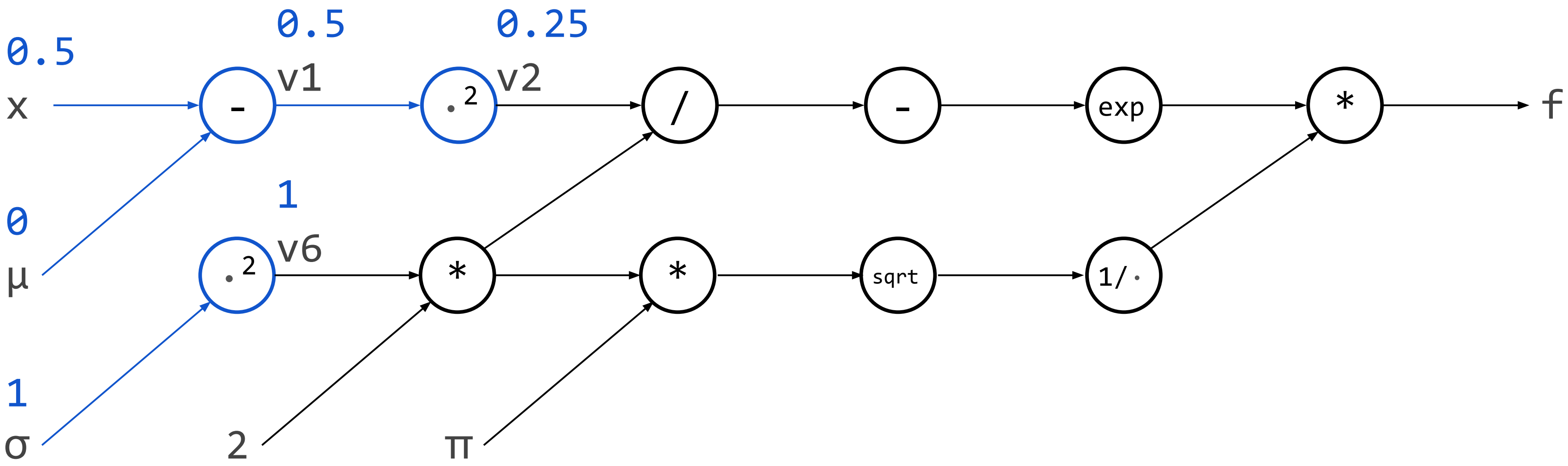




Normal PDF

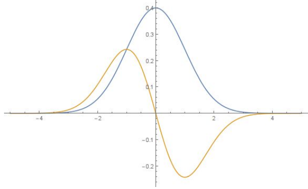
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

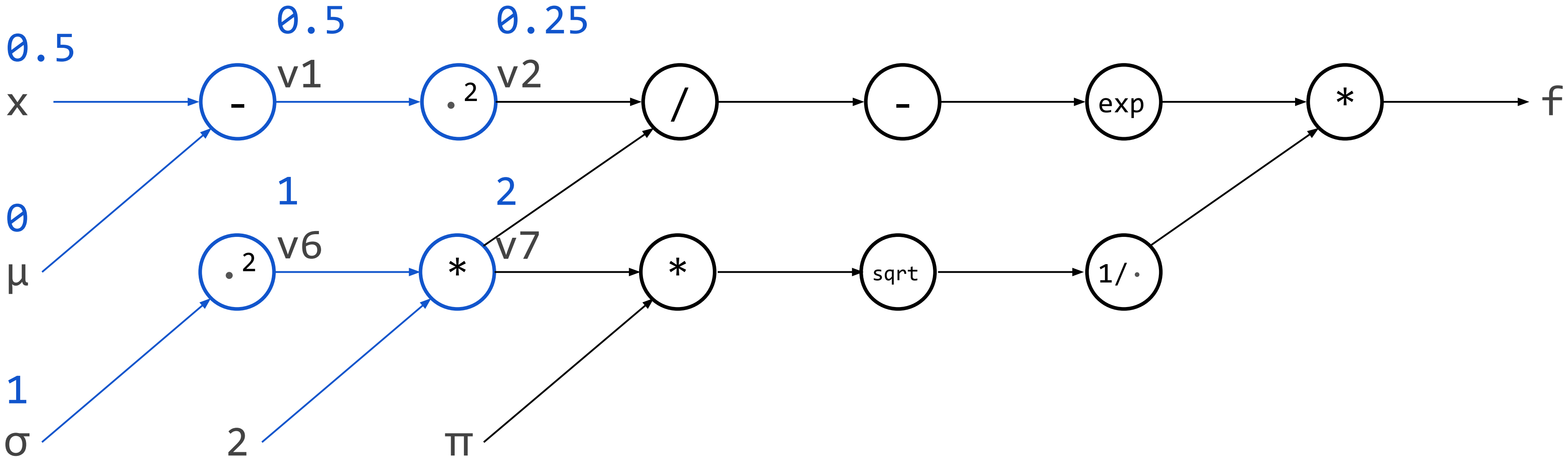




Normal PDF

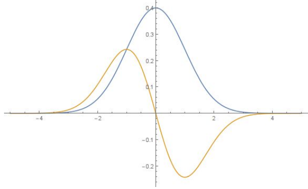
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

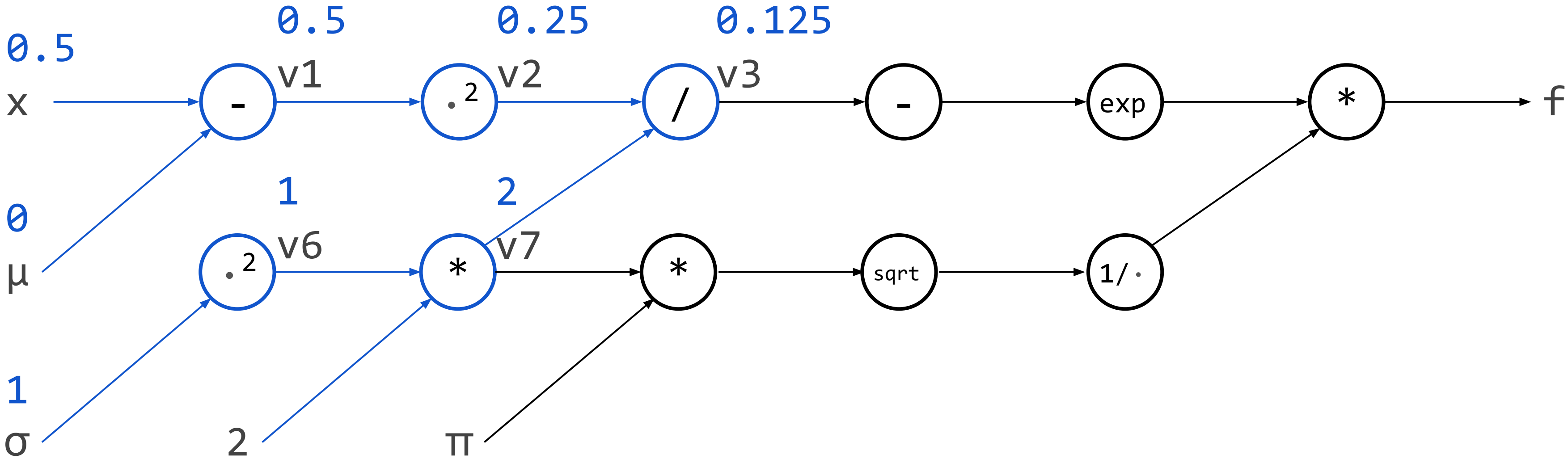




Normal PDF

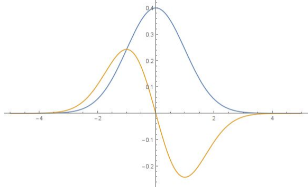
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

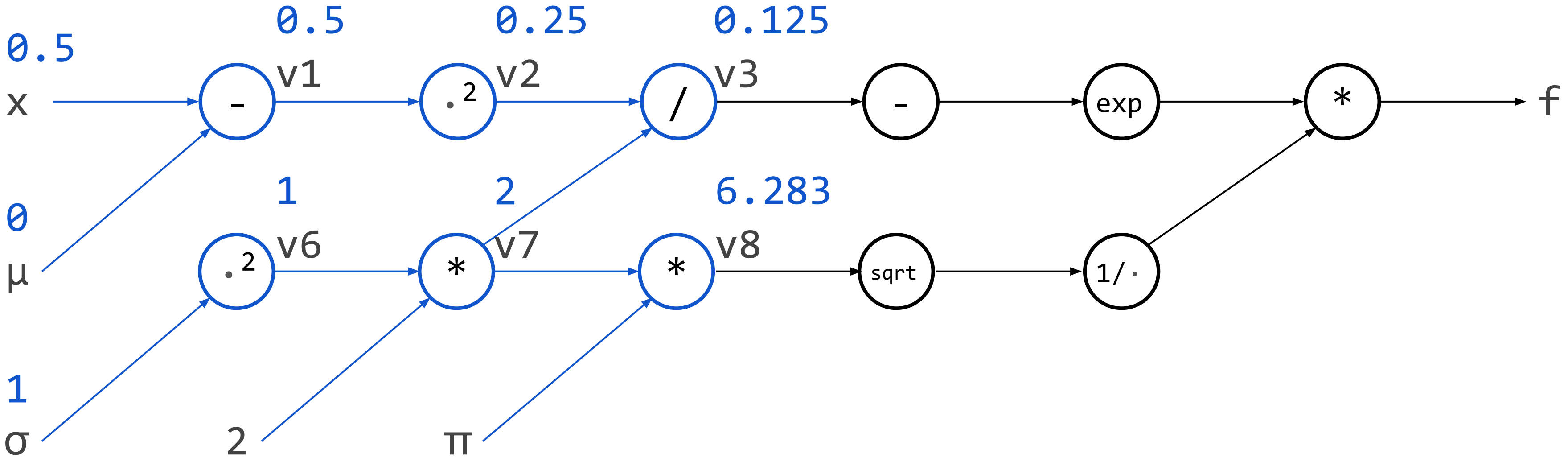




Normal PDF

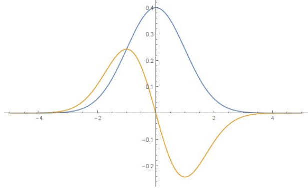
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

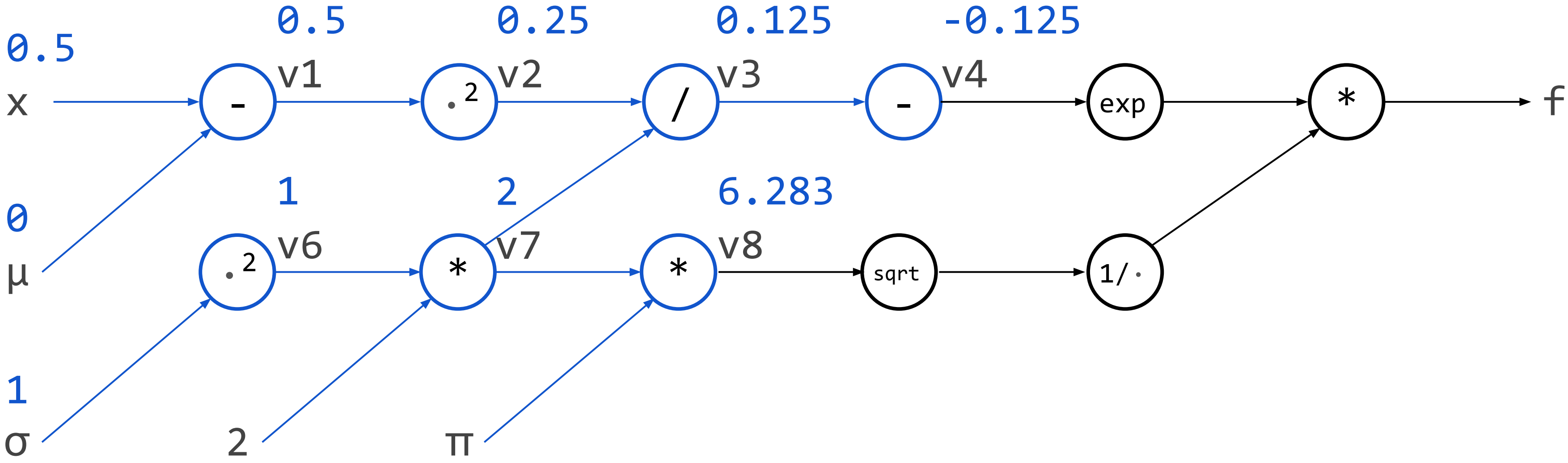




Normal PDF

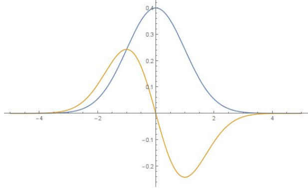
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

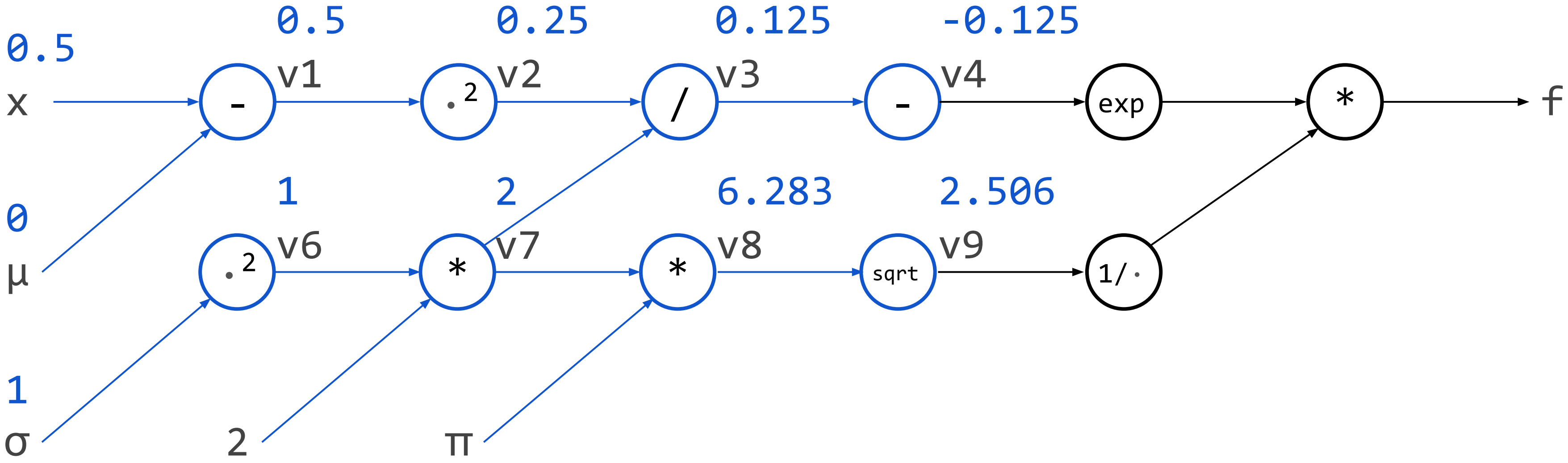




Normal PDF

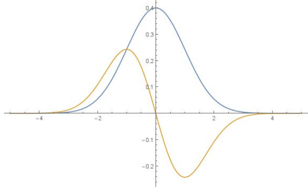
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

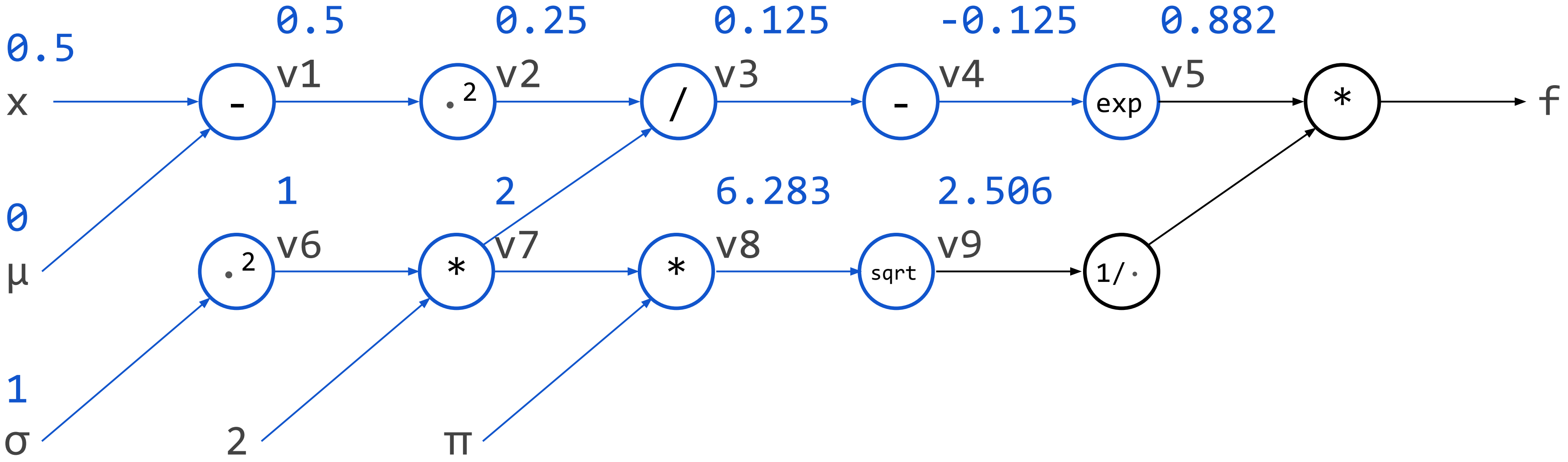




Normal PDF

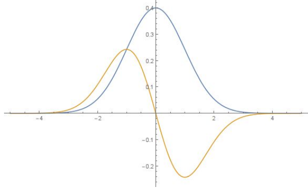
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

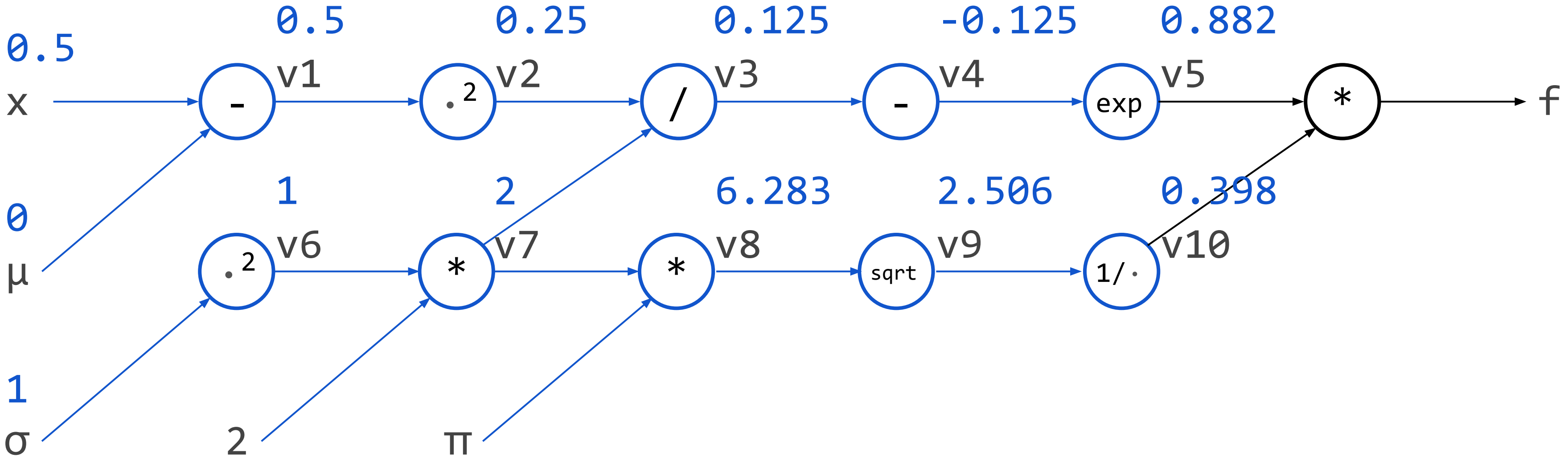




Normal PDF

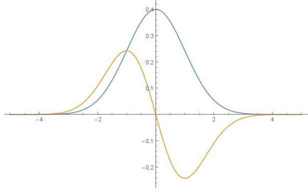
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

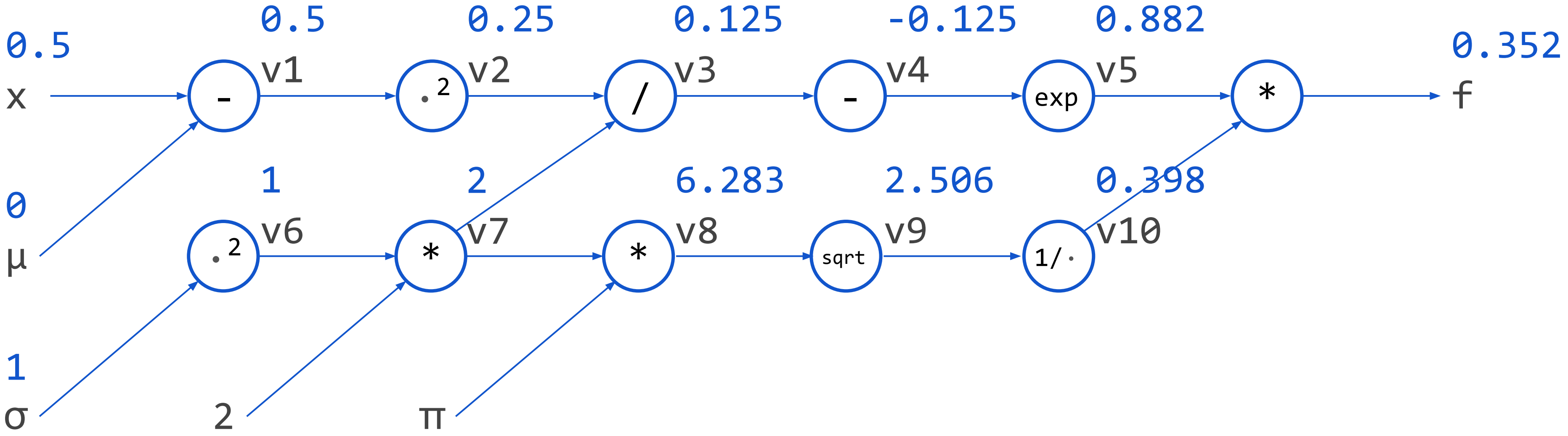




Normal PDF

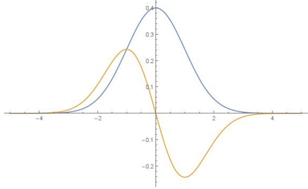
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

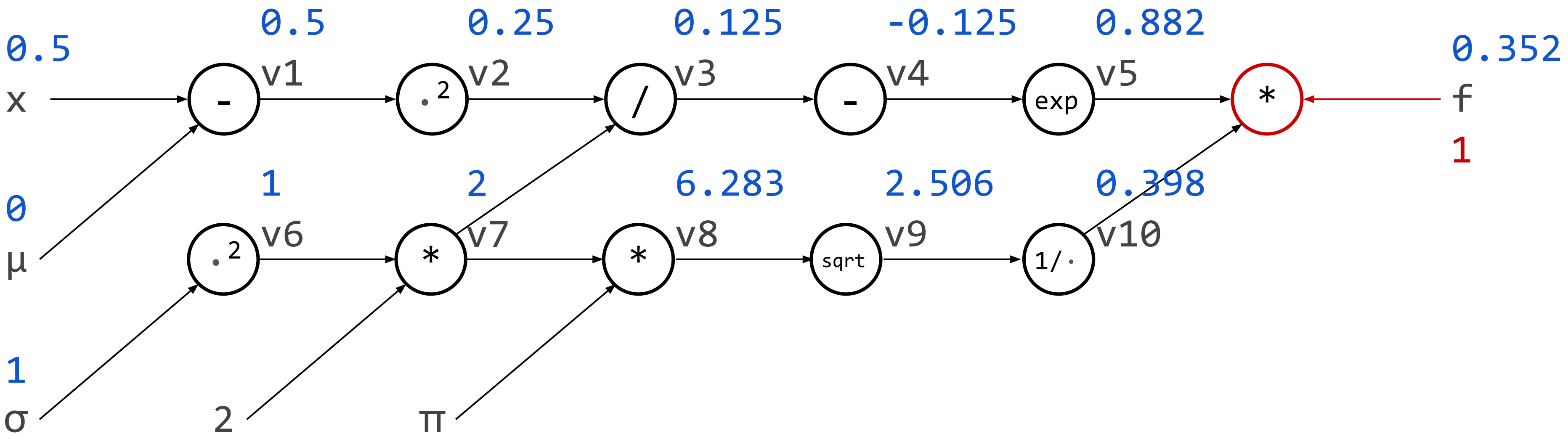




Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

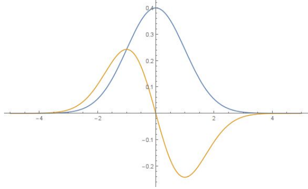


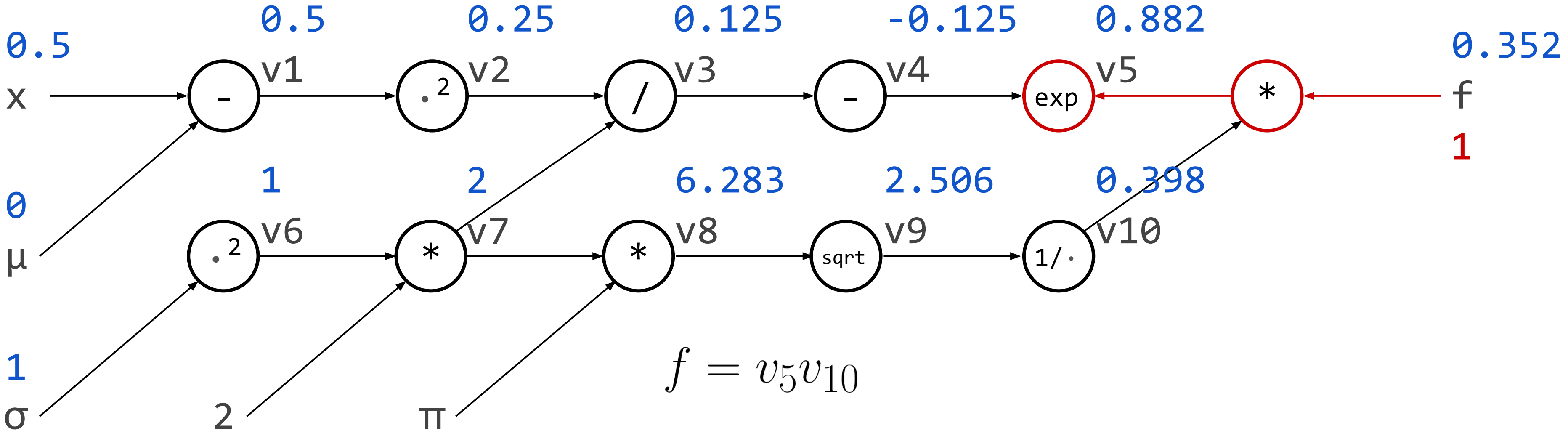


$$\frac{\partial f}{\partial f} = 1$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

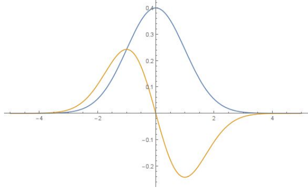


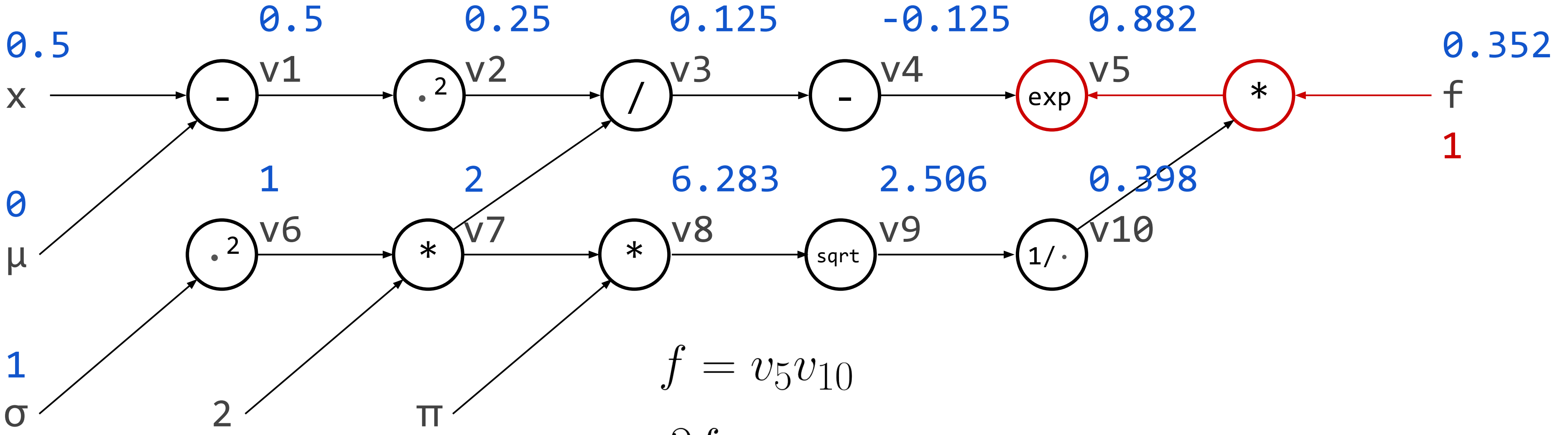


$$f = v_5 v_{10}$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



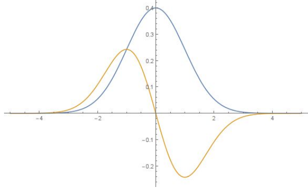


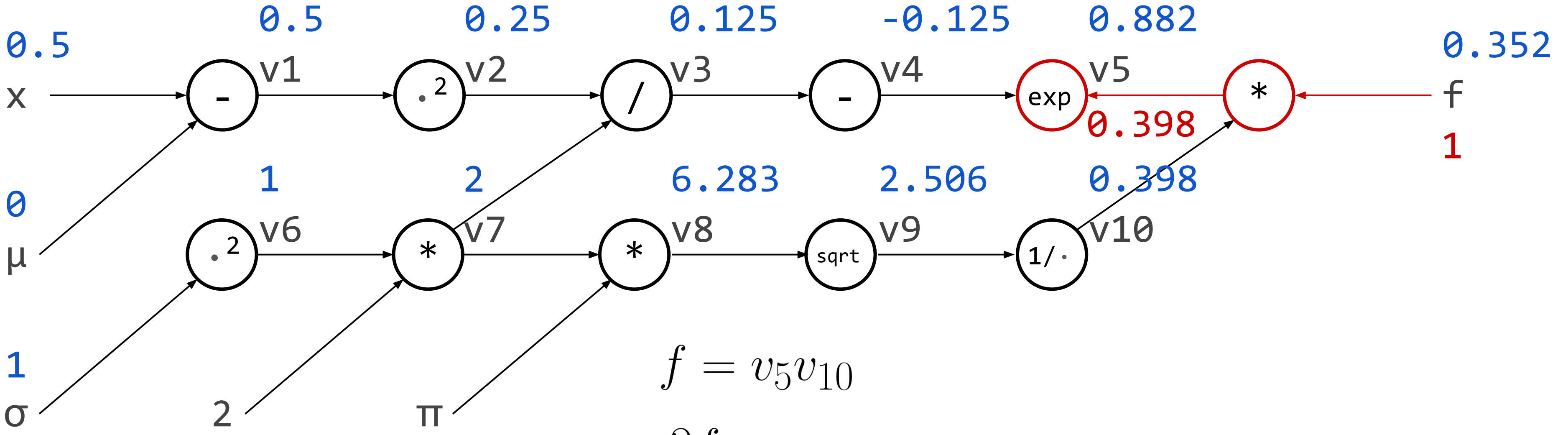
$$f = v_5 v_{10}$$

$$\frac{\partial f}{\partial v_5} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



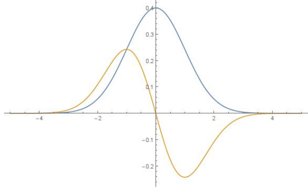


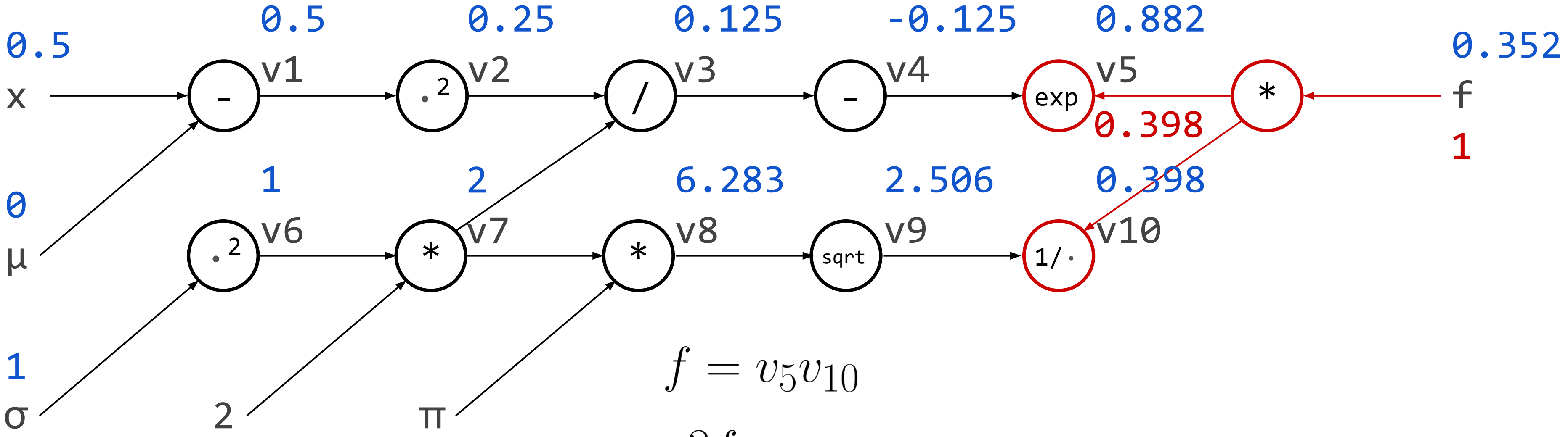
$$f = v_5 v_{10}$$

$$\frac{\partial f}{\partial v_5} = v_{10}$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



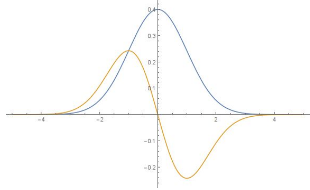


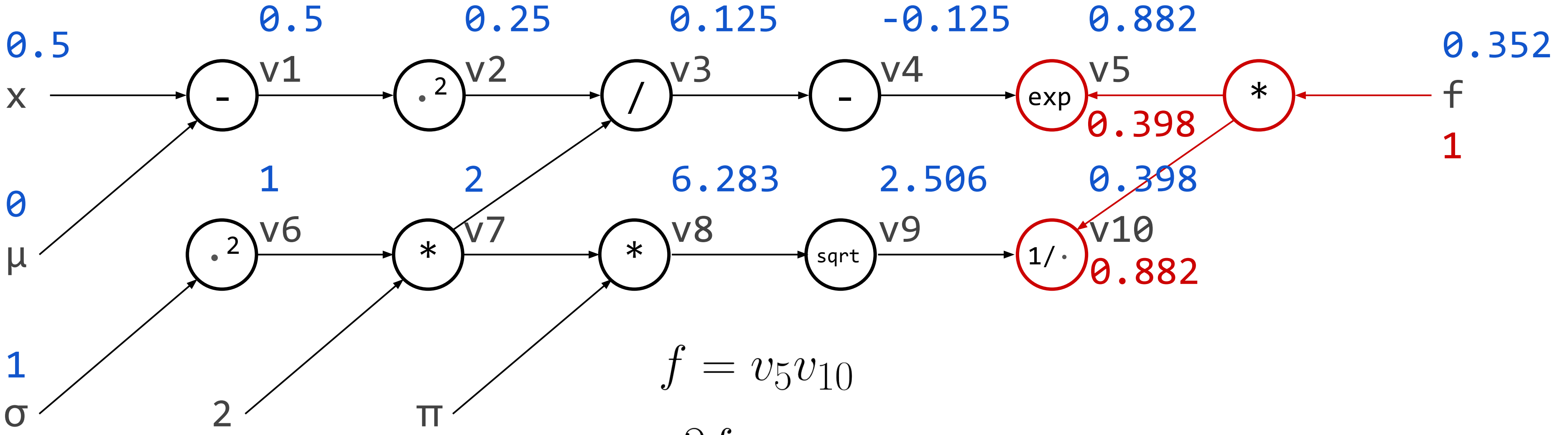
$$f = v_5 v_{10}$$

$$\frac{\partial f}{\partial v_{10}} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



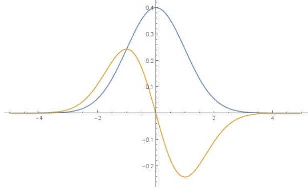


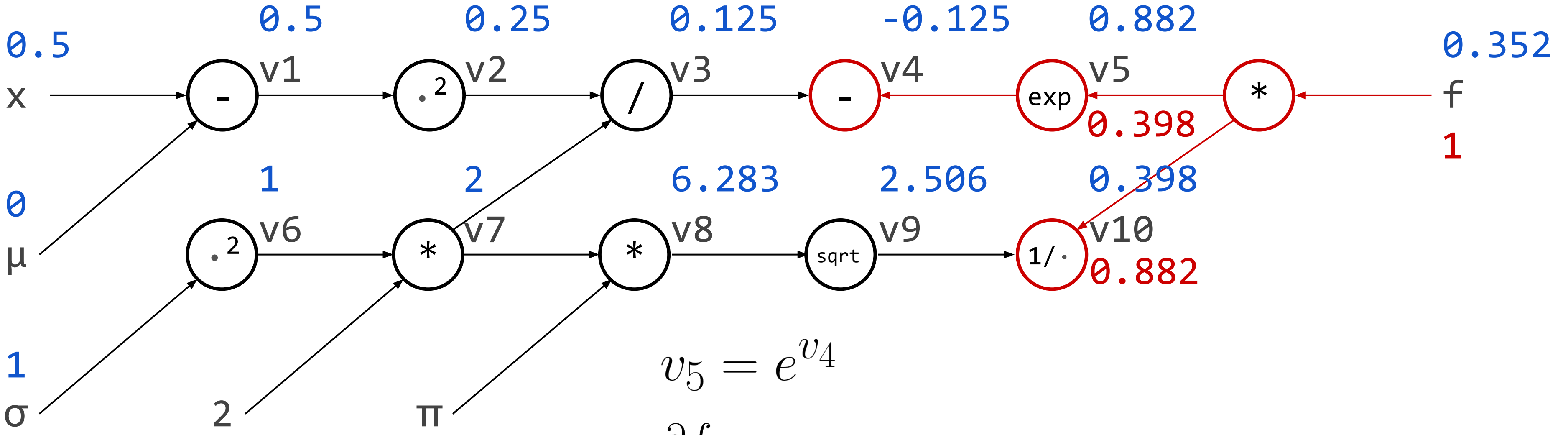
$$f = v_5 v_{10}$$

$$\frac{\partial f}{\partial v_{10}} = v_5$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



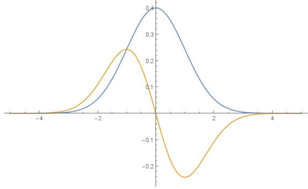


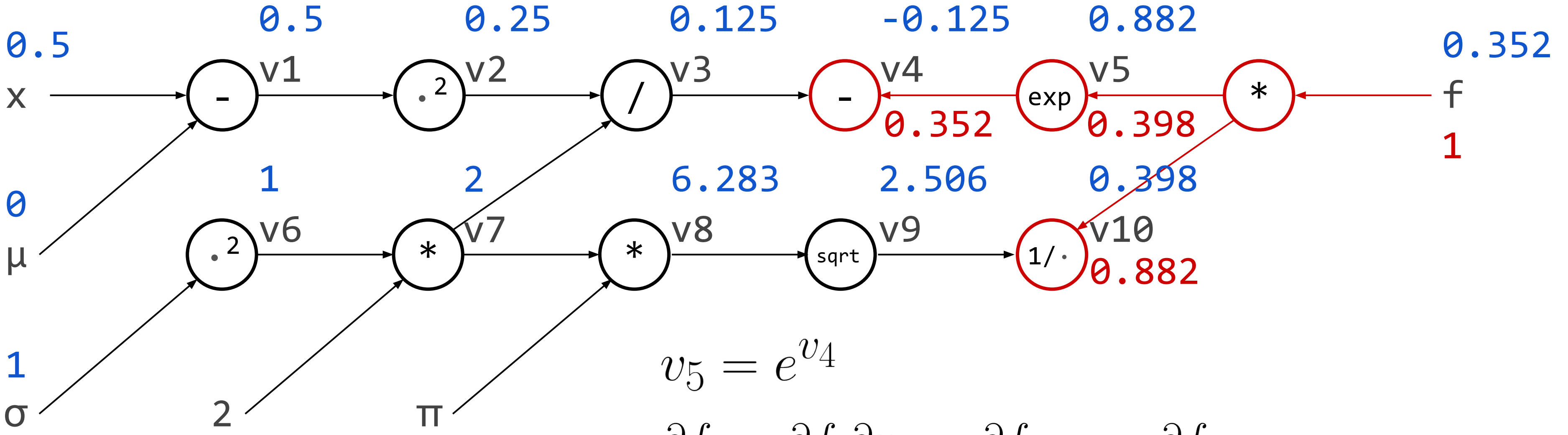
$$v_5 = e^{v_4}$$

$$\frac{\partial f}{\partial v_4} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



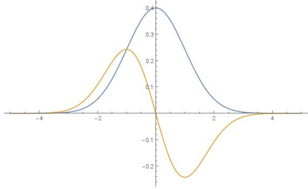


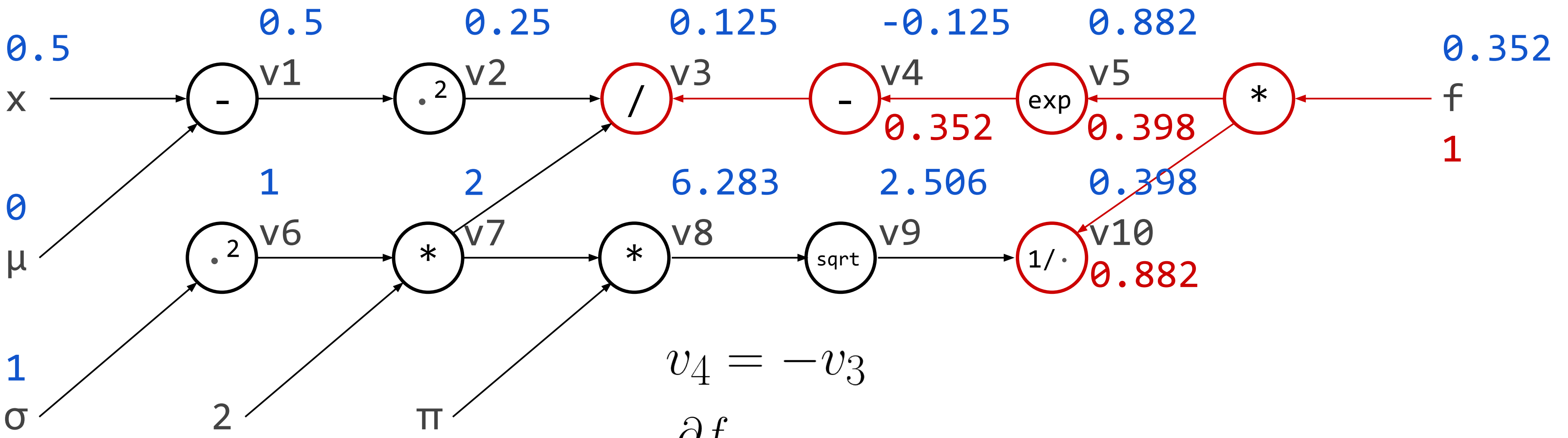
$$v_5 = e^{v_4}$$

$$\frac{\partial f}{\partial v_4} = \frac{\partial f}{\partial v_5}\frac{\partial v_5}{\partial v_4} = \frac{\partial f}{\partial v_5} e^{v_4} = \frac{\partial f}{\partial v_5} v_5$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



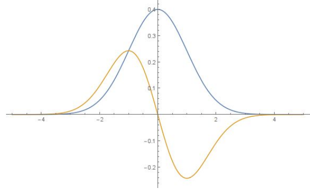


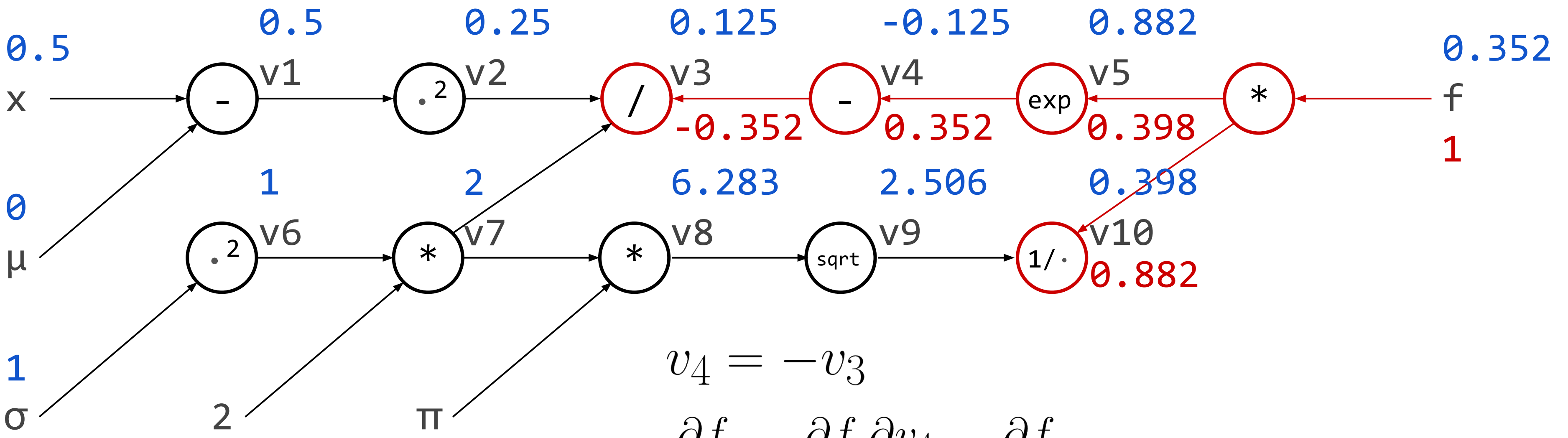
$$v_4 = -v_3$$

$$\frac{\partial f}{\partial v_3} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



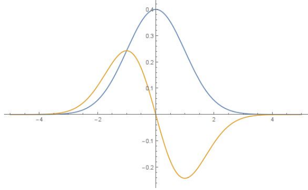


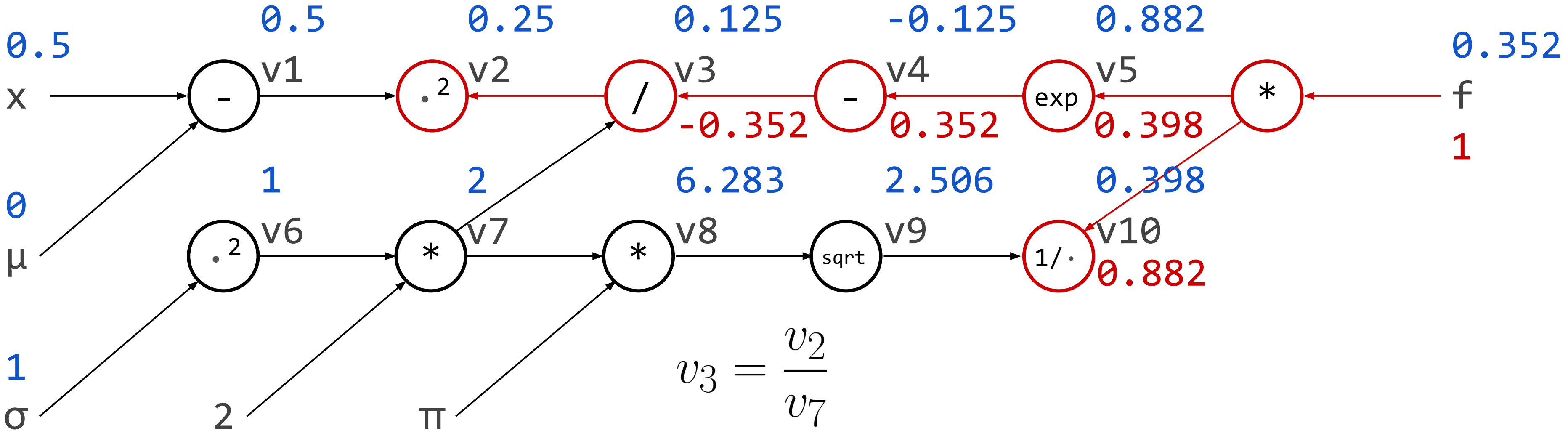
$$v_4 = -v_3$$

$$\frac{\partial f}{\partial v_3} = \frac{\partial f}{\partial v_4}\frac{\partial v_4}{\partial v_3} = \frac{\partial f}{\partial v_4}(-1)$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



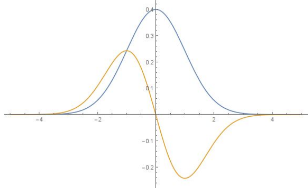


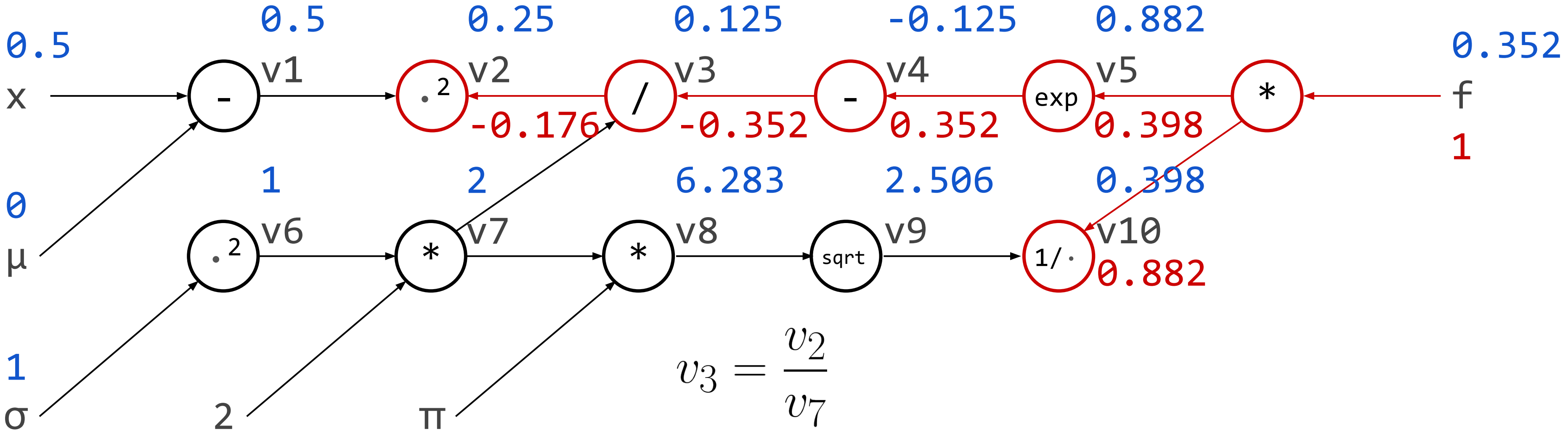
$$v_3 = \frac{v_2}{v_7}$$

$$\frac{\partial f}{\partial v_2} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



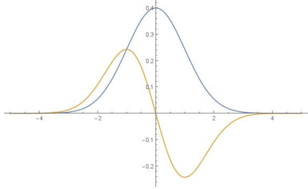


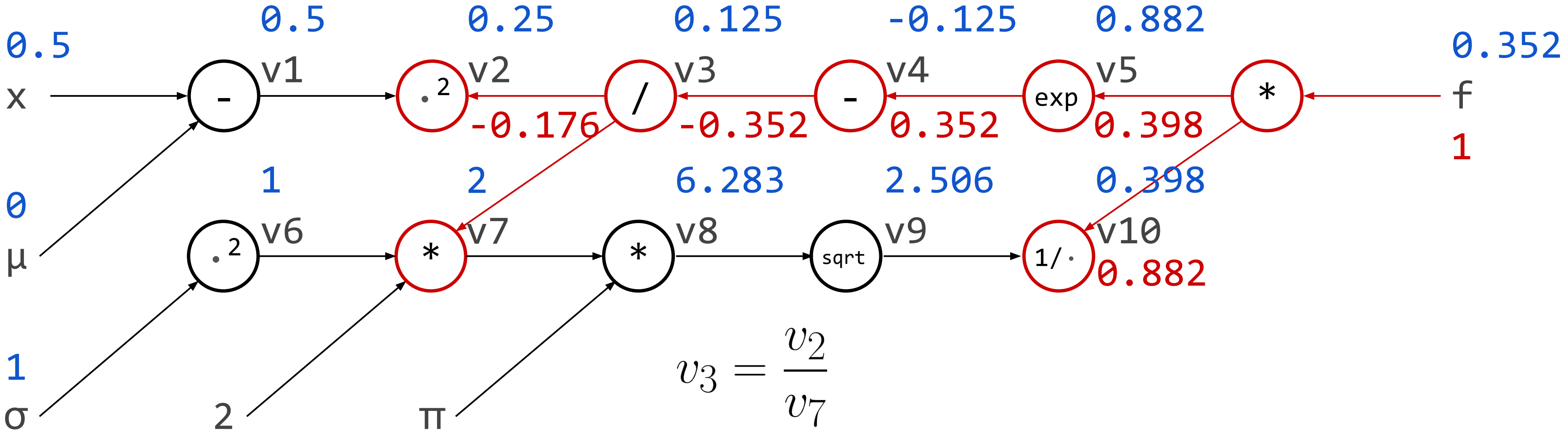
$$v_3 = \frac{v_2}{v_7}$$

$$\frac{\partial f}{\partial v_2} = \frac{\partial f}{\partial v_3}\frac{\partial v_3}{\partial v_2} = \frac{\partial f}{\partial v_3}\frac{1}{v_7}$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



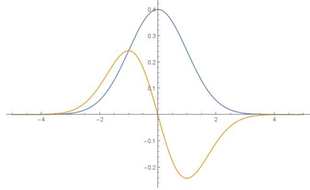


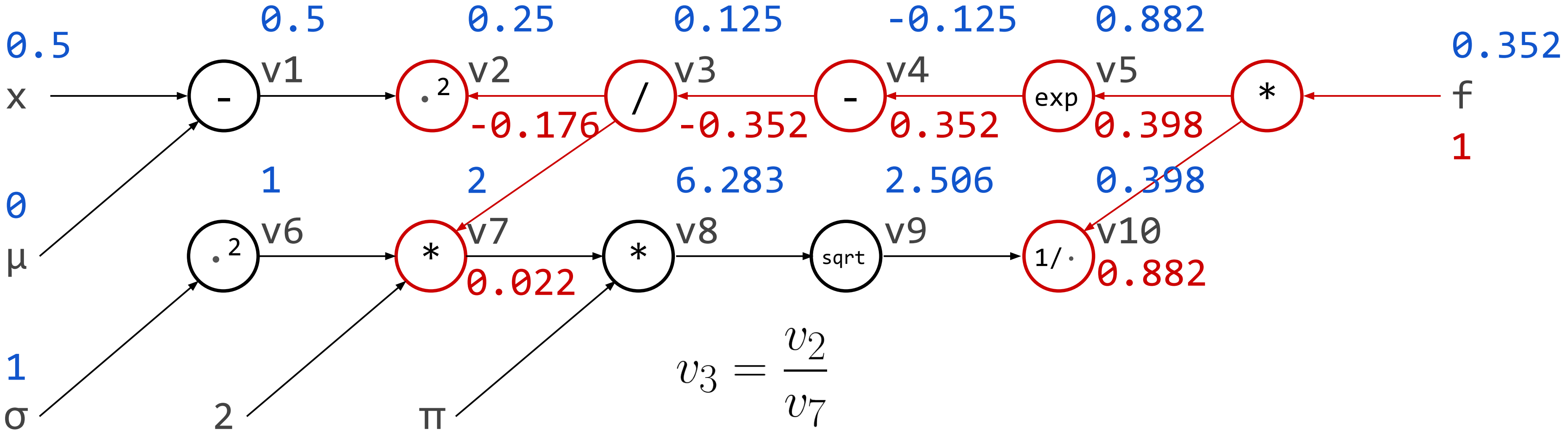
$$v_3 = \frac{v_2}{v_7}$$

$$\frac{\partial f}{\partial v_7} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



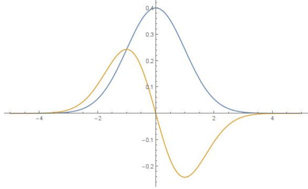


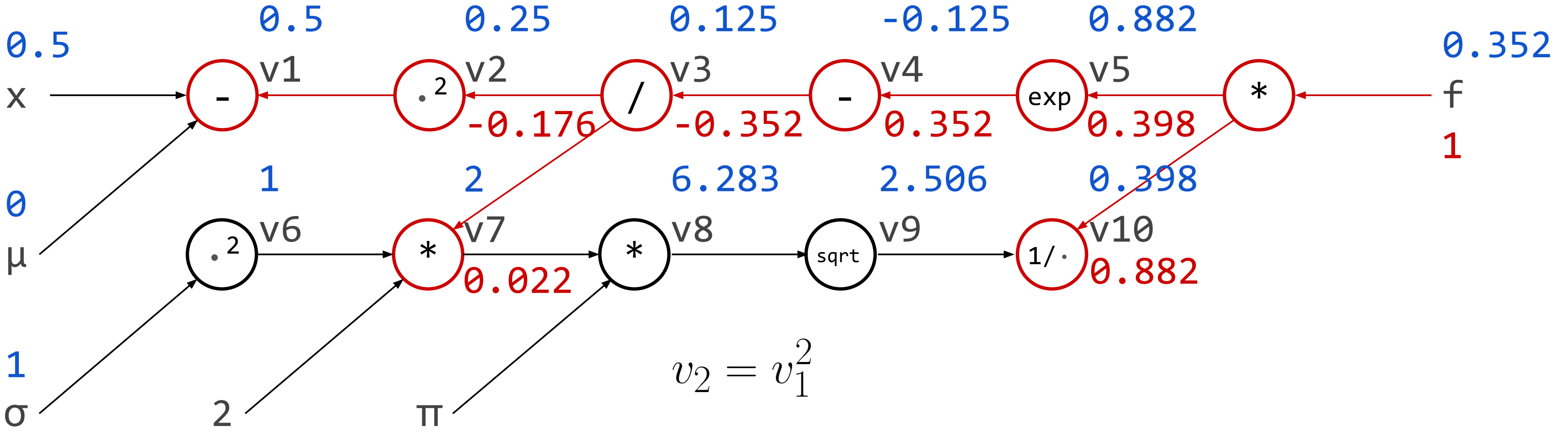
$$v_3 = \frac{v_2}{v_7}$$

$$\frac{\partial f}{\partial v_7} = \frac{\partial f}{\partial v_3}\frac{\partial v_3}{\partial v_7} + \cdots = \frac{\partial f}{\partial v_3}\frac{-v_2}{v_7^2} + \dots$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



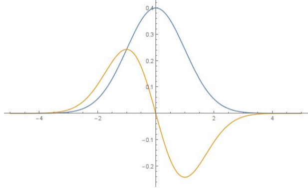


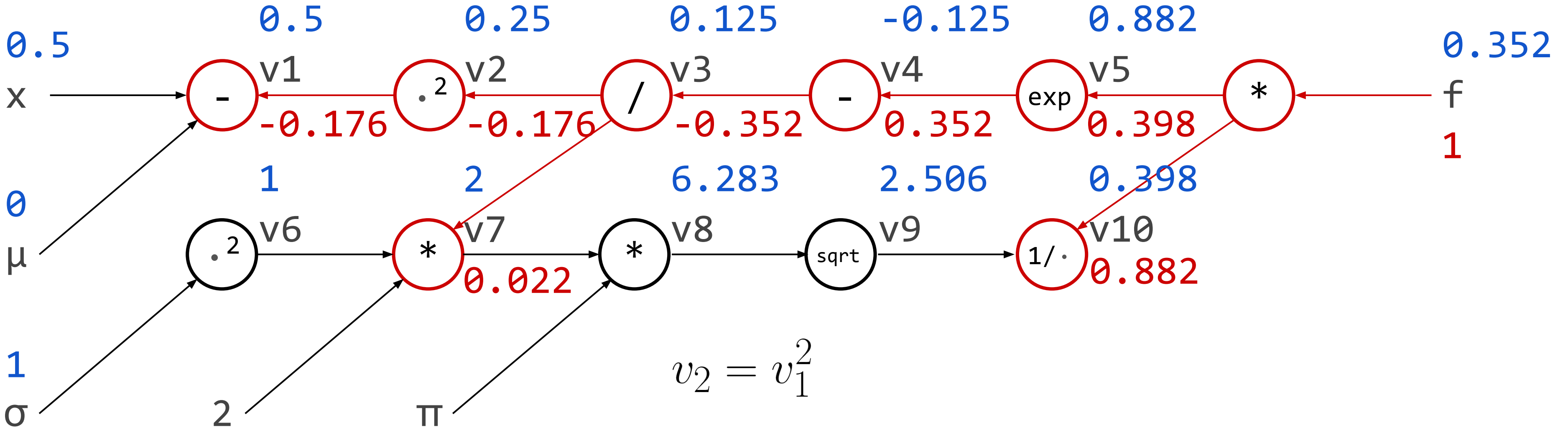
$$v_2 = v_1^2$$

$$\frac{\partial f}{\partial v_1} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



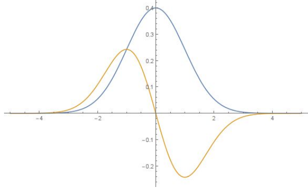


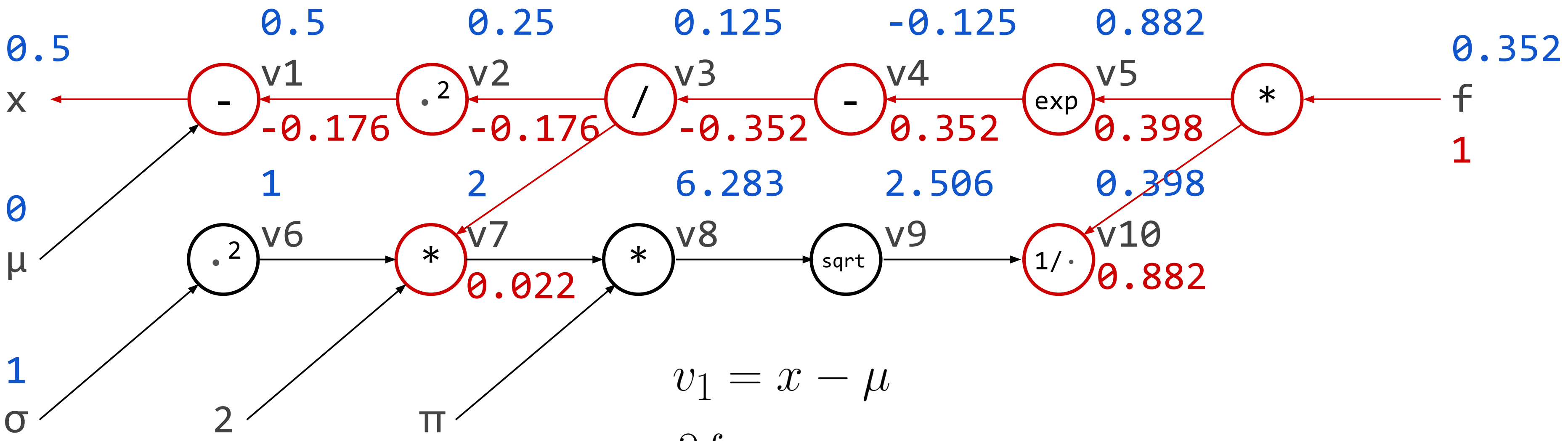
$$v_2 = v_1^2$$

$$\frac{\partial f}{\partial v_1} = \frac{\partial f}{\partial v_2}\frac{\partial v_2}{\partial v_1} = \frac{\partial f}{\partial v_2} 2v_1$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



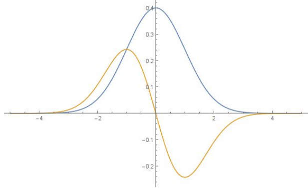


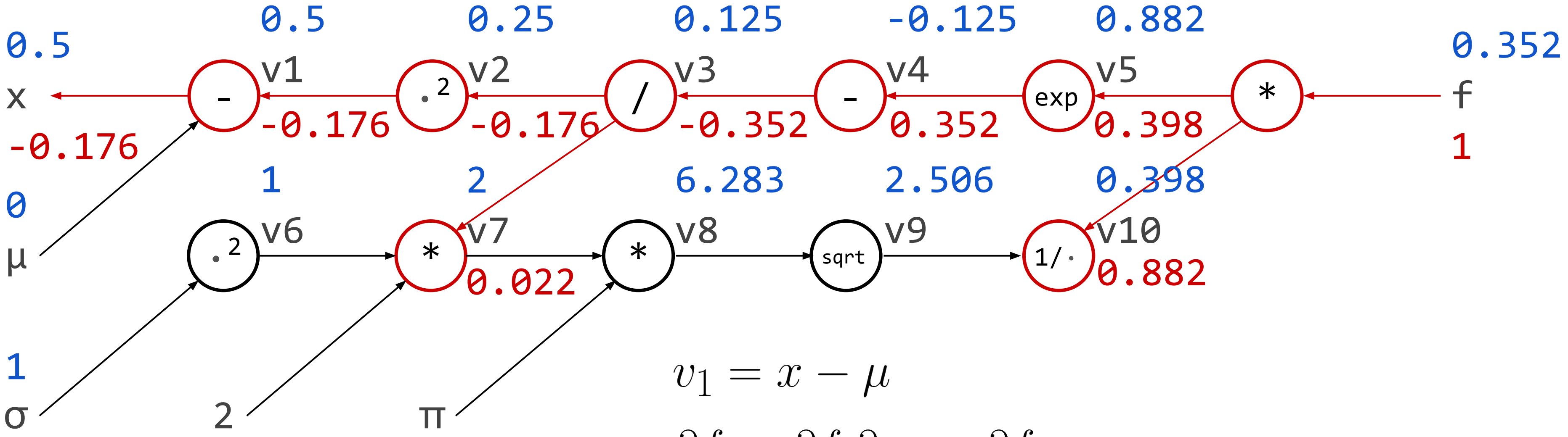
$$v_1 = x - \mu$$

$$\frac{\partial f}{\partial x} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



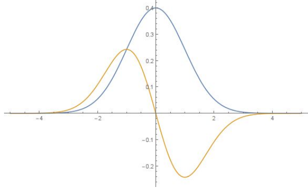


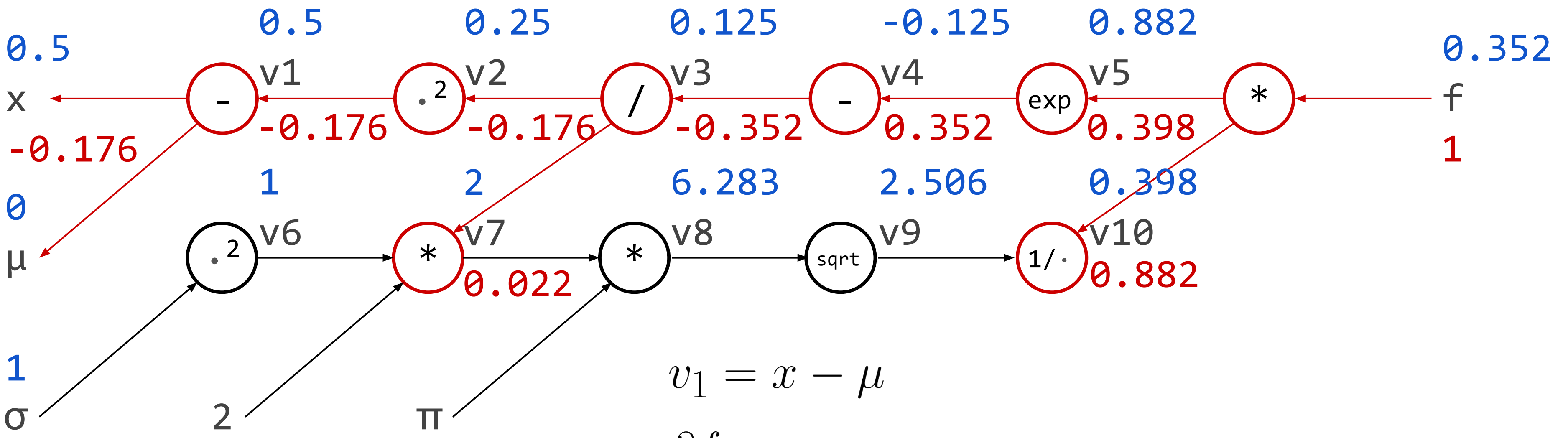
$$v_1 = x - \mu$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial v_1}\frac{\partial v_1}{\partial x} = \frac{\partial f}{\partial v_1} 1$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



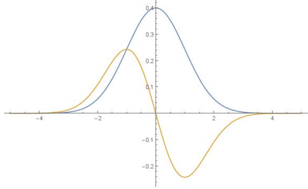


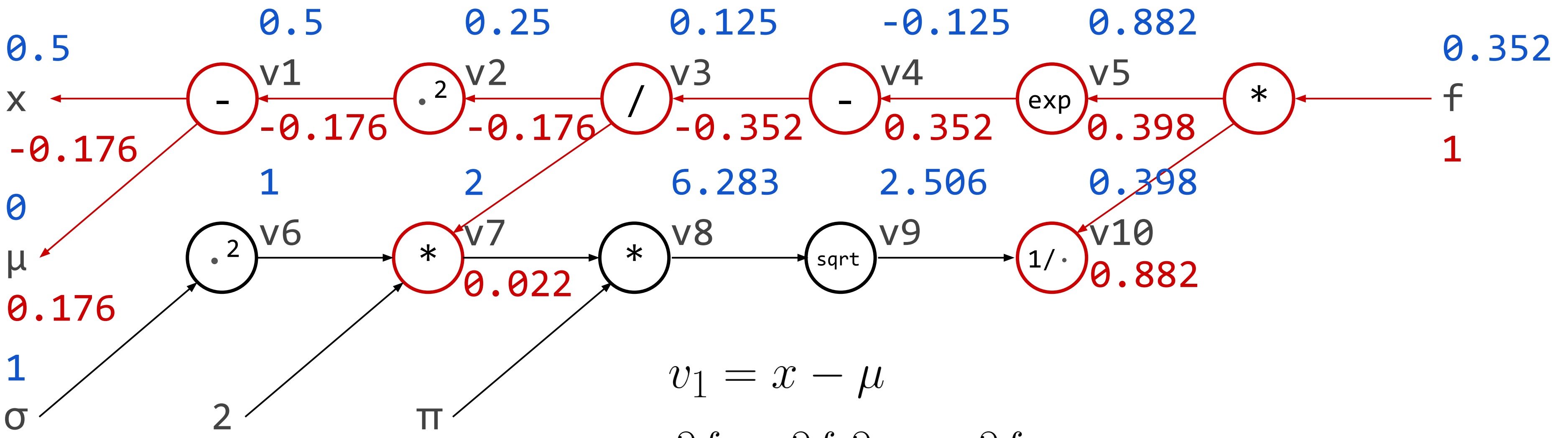
$$v_1 = x - \mu$$

$$\frac{\partial f}{\partial \mu} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



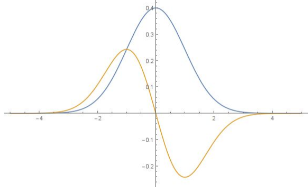


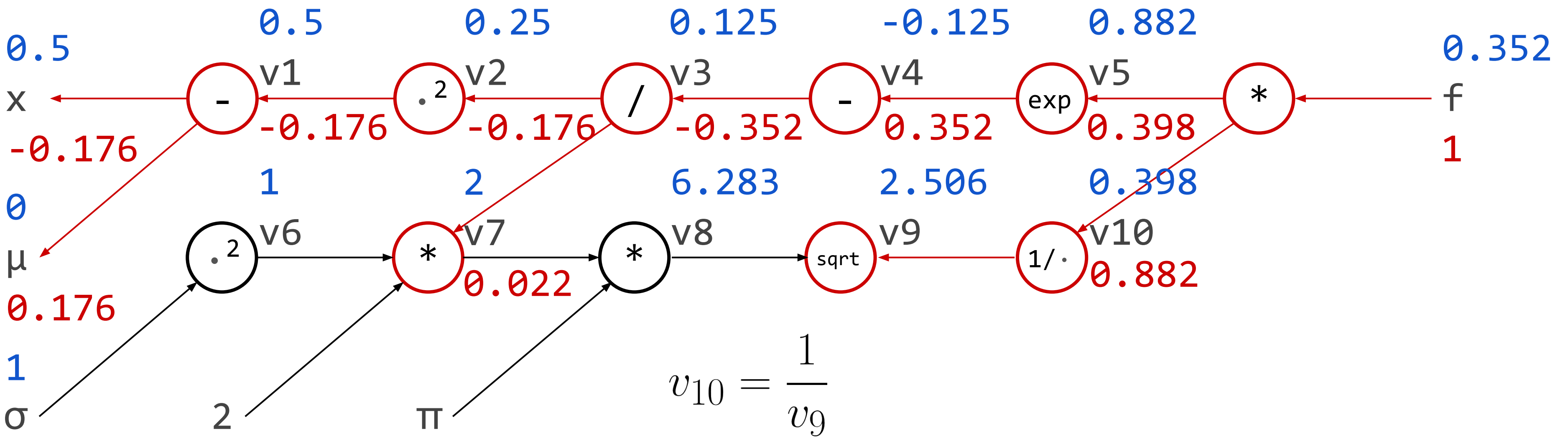
$$v_1 = x - \mu$$

$$\frac{\partial f}{\partial \mu} = \frac{\partial f}{\partial v_1}\frac{\partial v_1}{\partial \mu} = \frac{\partial f}{\partial v_1}(-1)$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



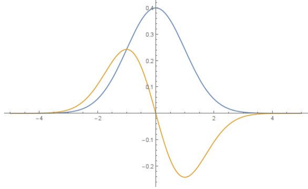


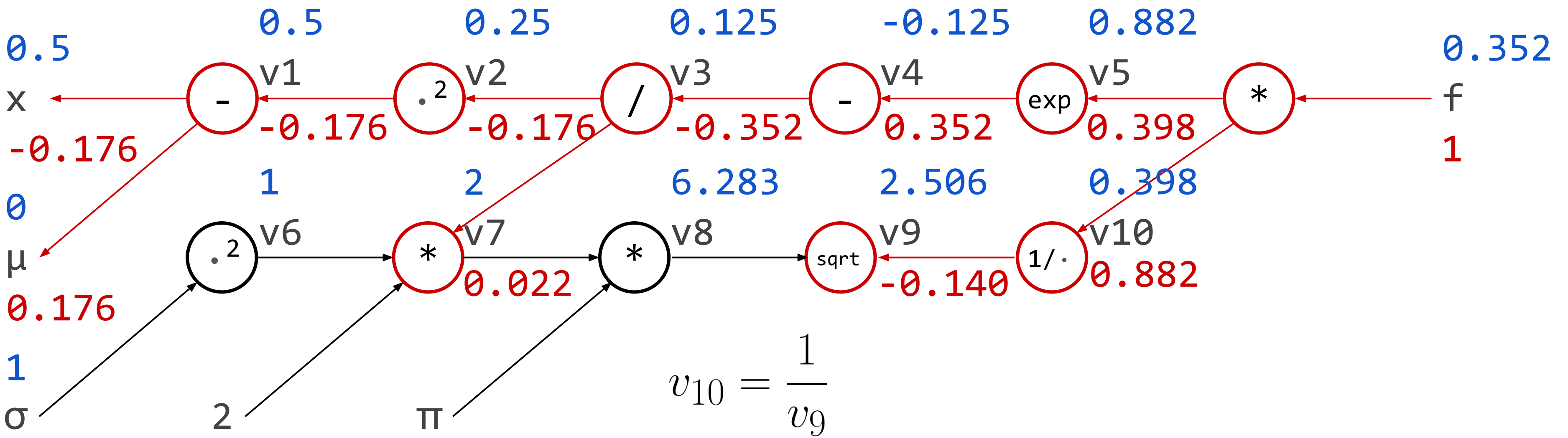
$$v_{10} = \frac{1}{v_9}$$

$$\frac{\partial f}{\partial v_9} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



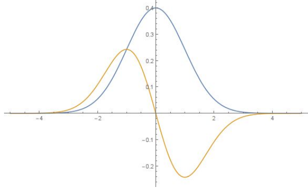


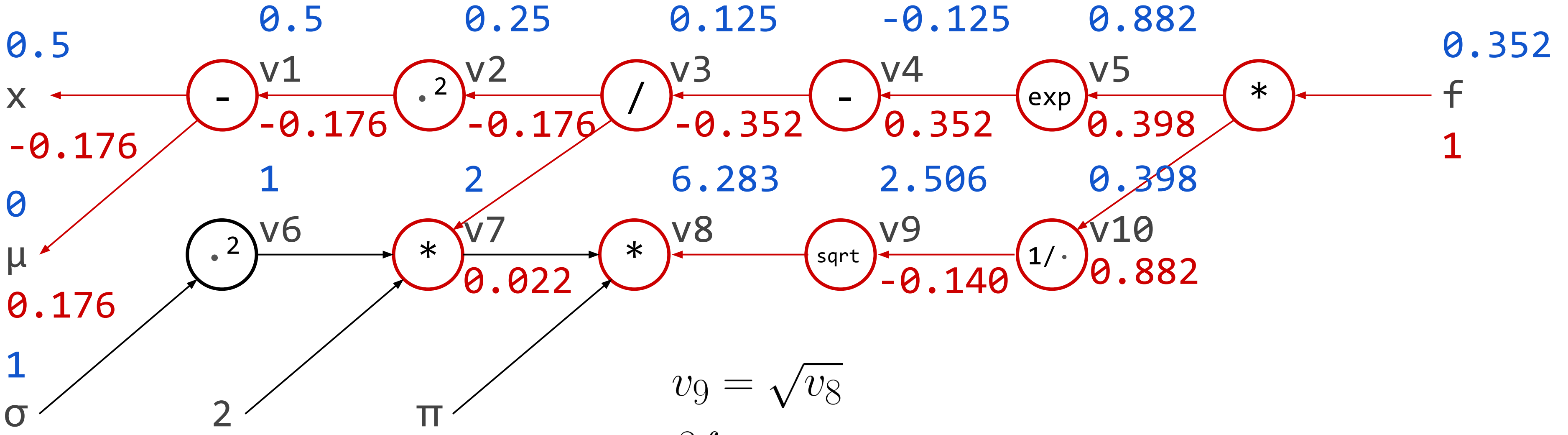
$$v_{10} = \frac{1}{v_9}$$

$$\frac{\partial f}{\partial v_9} = \frac{\partial f}{\partial v_{10}} \frac{\partial v_{10}}{\partial v_9} = \frac{\partial f}{\partial v_{10}} \frac{-1}{v_9^2}$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



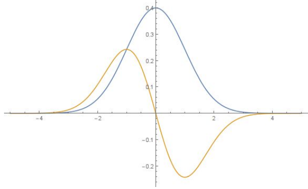


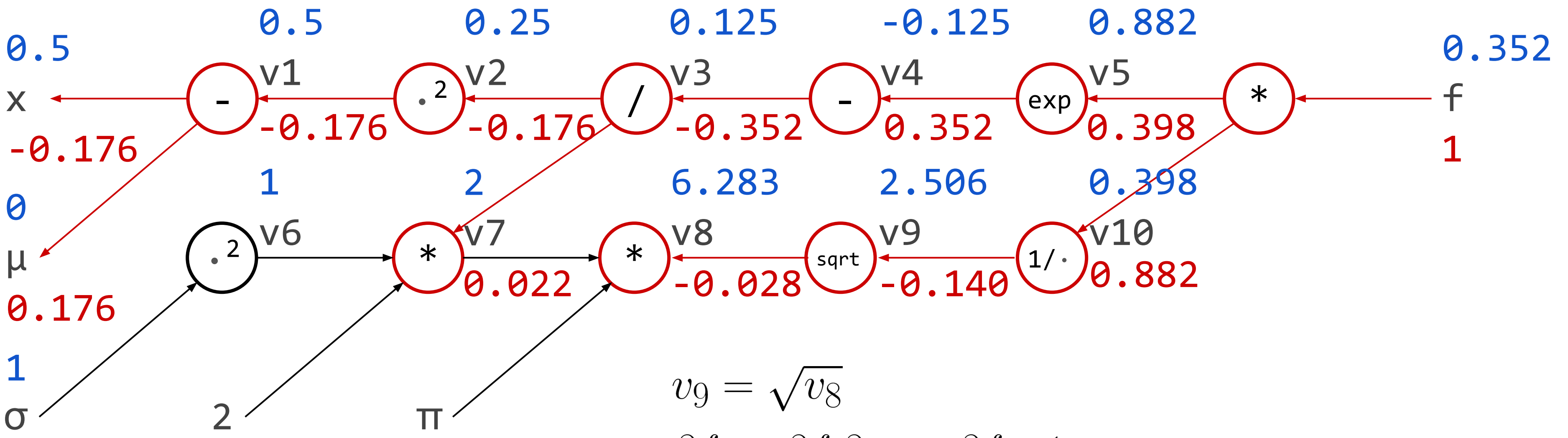
$$v_9 = \sqrt{v_8}$$

$$\frac{\partial f}{\partial v_8} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



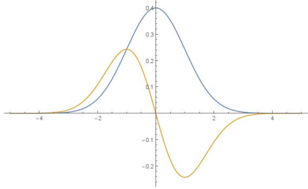


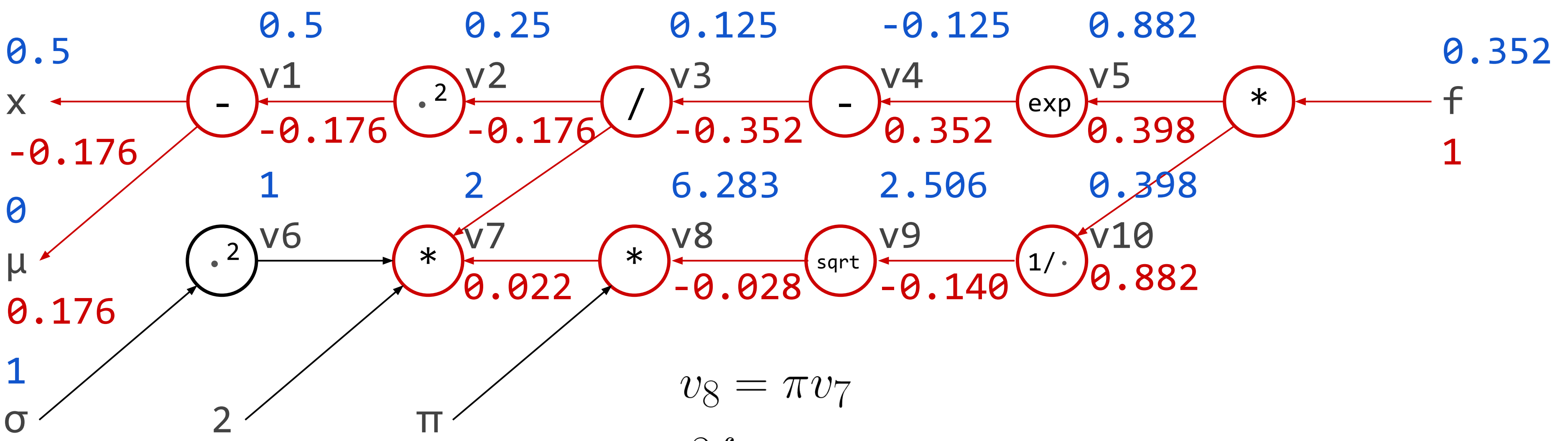
$$v_9 = \sqrt{v_8}$$

$$\frac{\partial f}{\partial v_8} = \frac{\partial f}{\partial v_9}\frac{\partial v_9}{\partial v_8} = \frac{\partial f}{\partial v_9}\frac{1}{2\sqrt{v_8}}$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



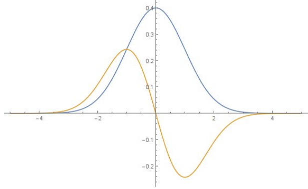


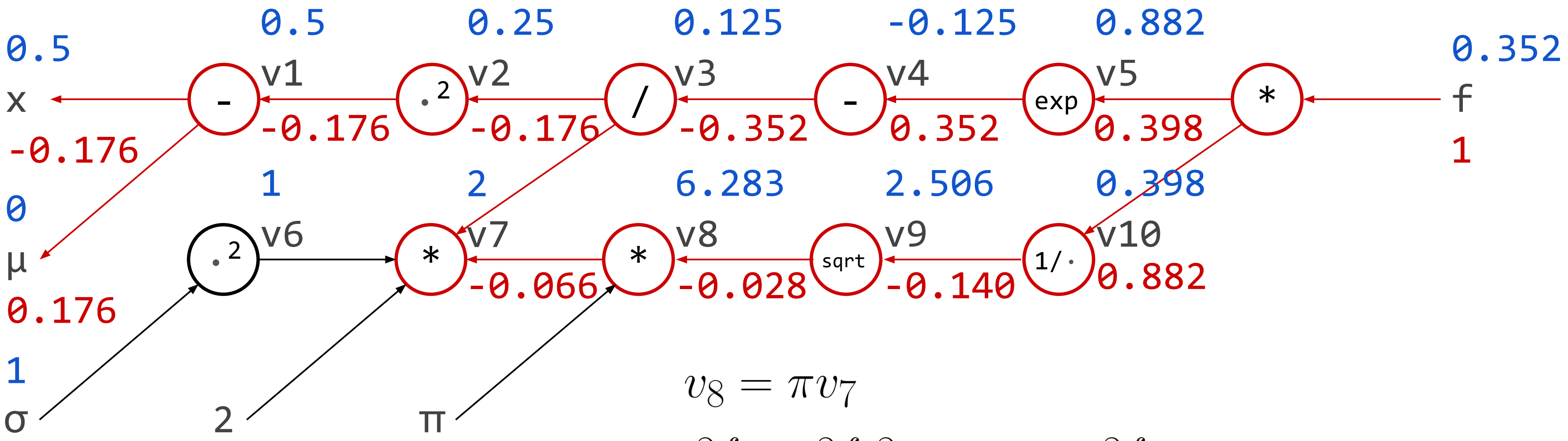
$$v_8 = \pi v_7$$

$$\frac{\partial f}{\partial v_7} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



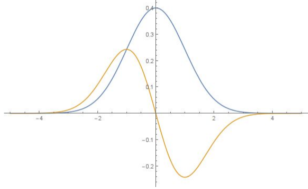


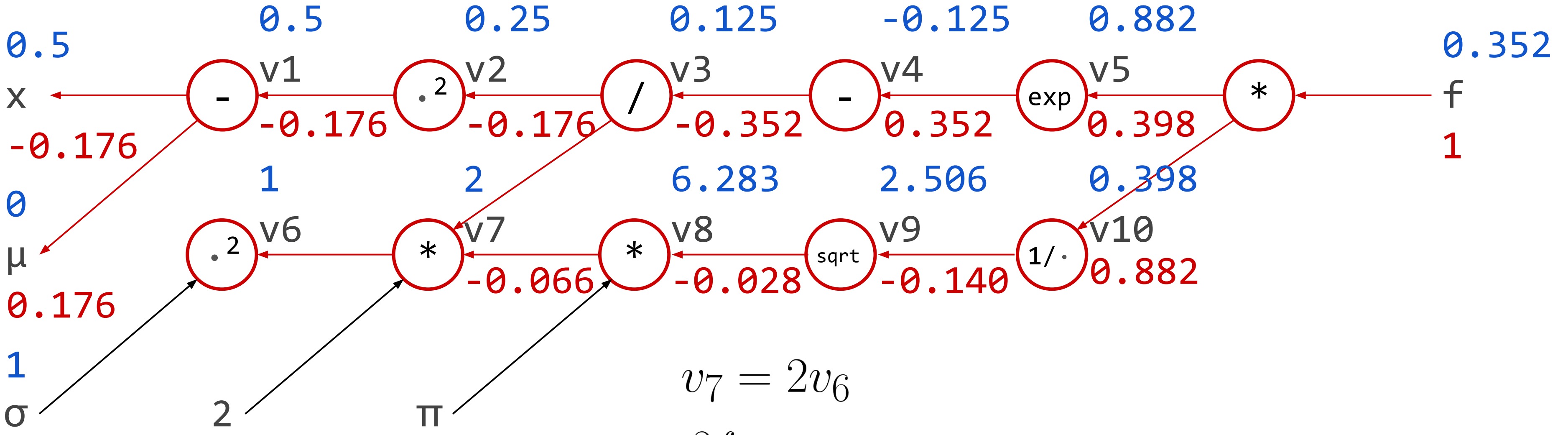
$$v_8 = \pi v_7$$

$$\frac{\partial f}{\partial v_7} = \frac{\partial f}{\partial v_8}\frac{\partial v_8}{\partial v_7} + \cdots = \frac{\partial f}{\partial v_8}\pi + \ldots$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



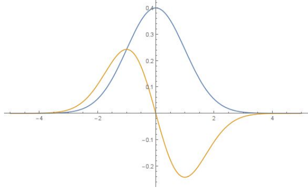


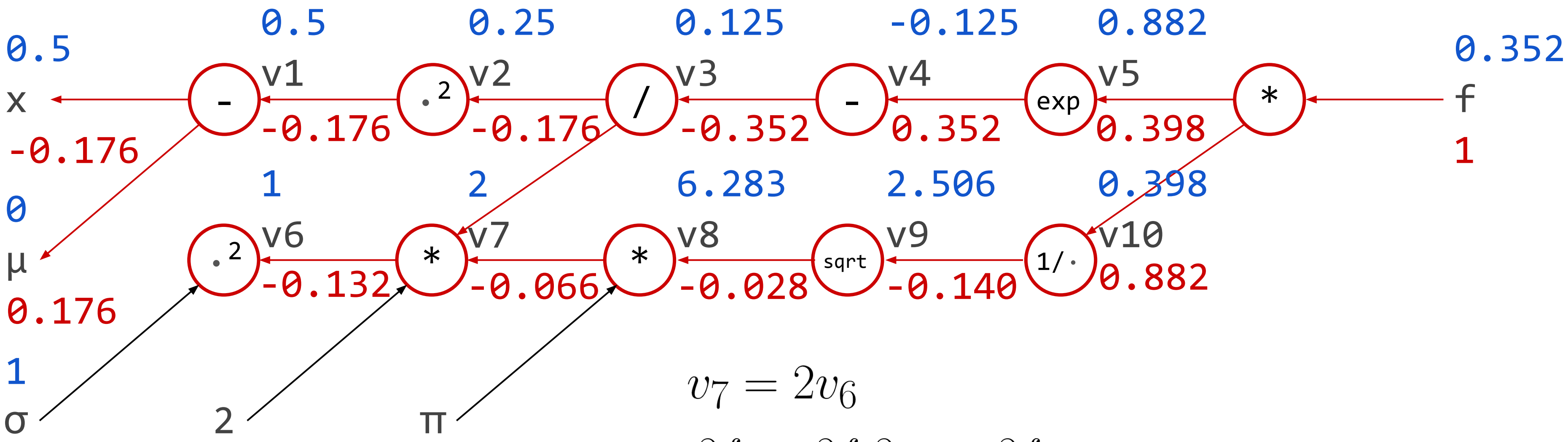
$$v_7 = 2v_6$$

$$\frac{\partial f}{\partial v_6} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



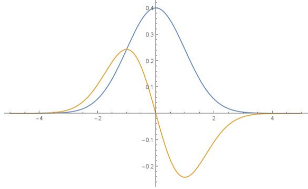


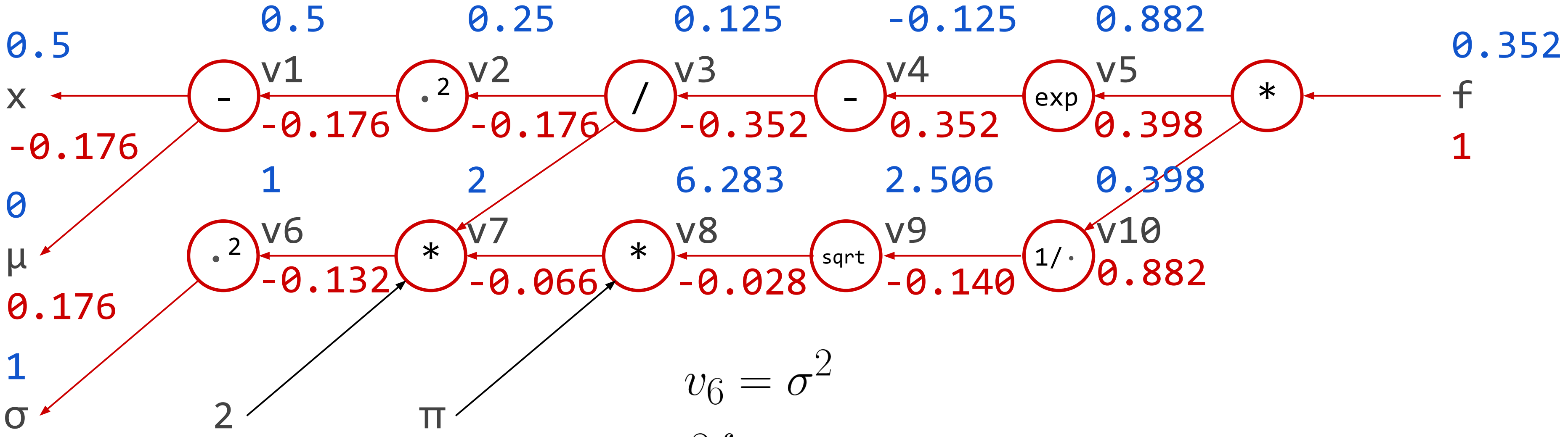
$$v_7 = 2v_6$$

$$\frac{\partial f}{\partial v_6} = \frac{\partial f}{\partial v_7}\frac{\partial v_7}{\partial v_6} = \frac{\partial f}{\partial v_7}2$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



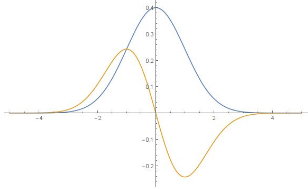


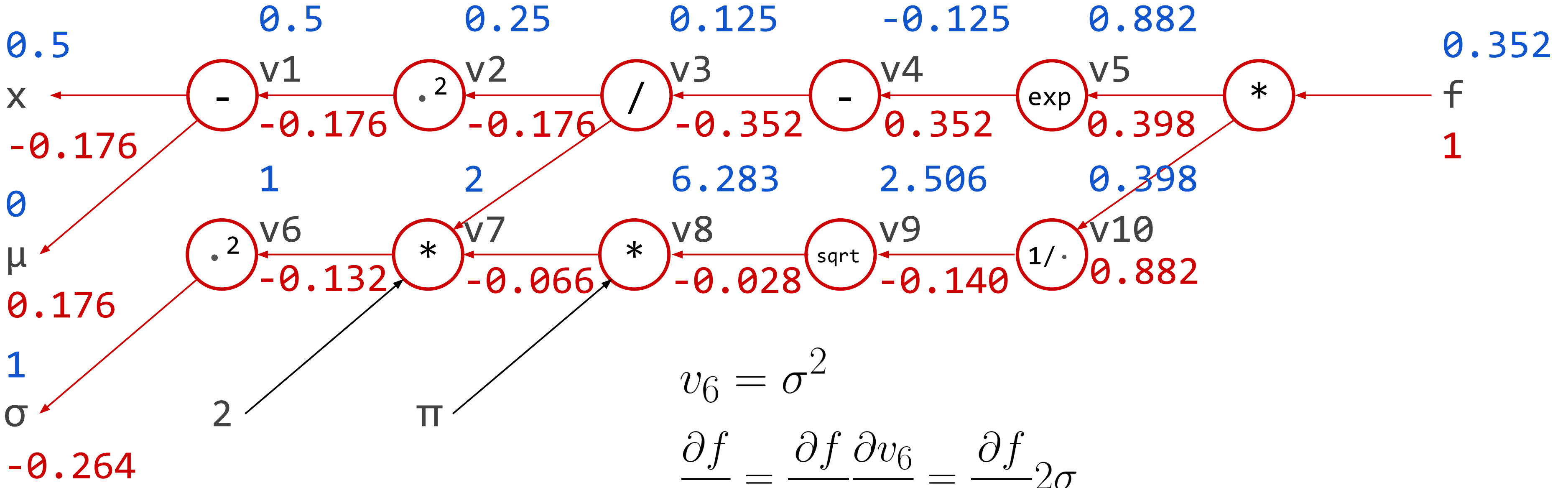
$$v_6 = \sigma^2$$

$$\frac{\partial f}{\partial \sigma} =$$

Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



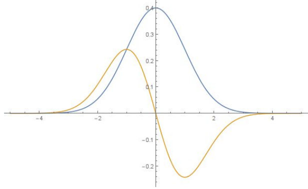


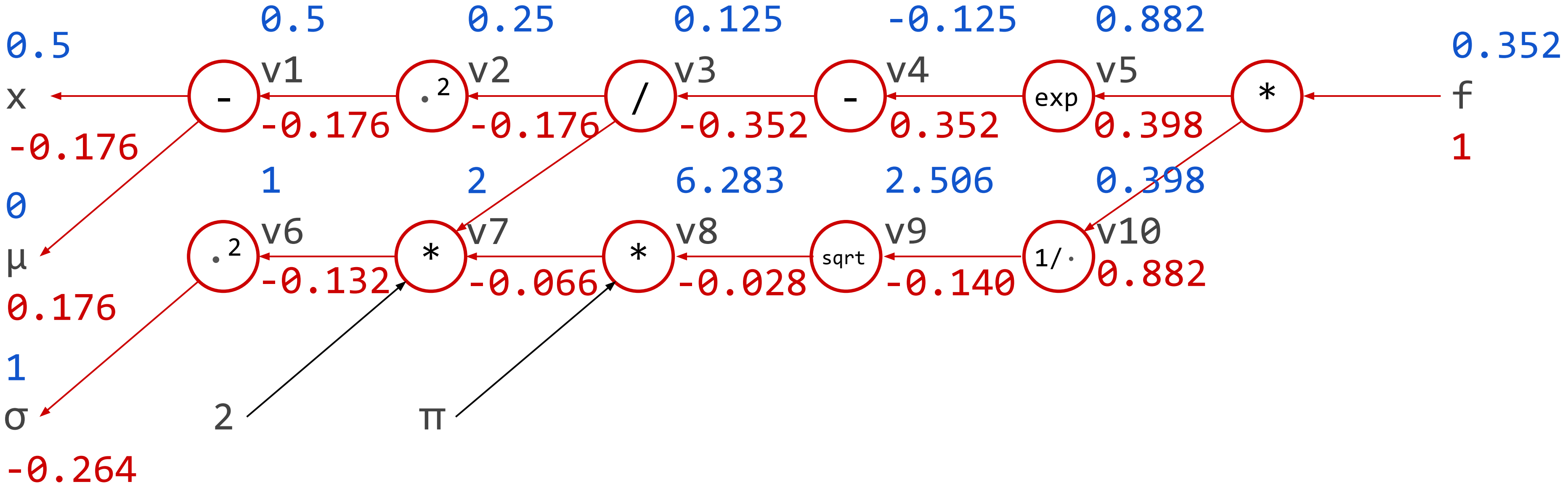
$$v_6 = \sigma^2$$

$$\frac{\partial f}{\partial \sigma} = \frac{\partial f}{\partial v_6}\frac{\partial v_6}{\partial \sigma} = \frac{\partial f}{\partial v_6} 2\sigma$$

Normal PDF

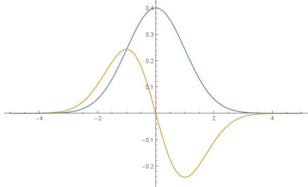
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



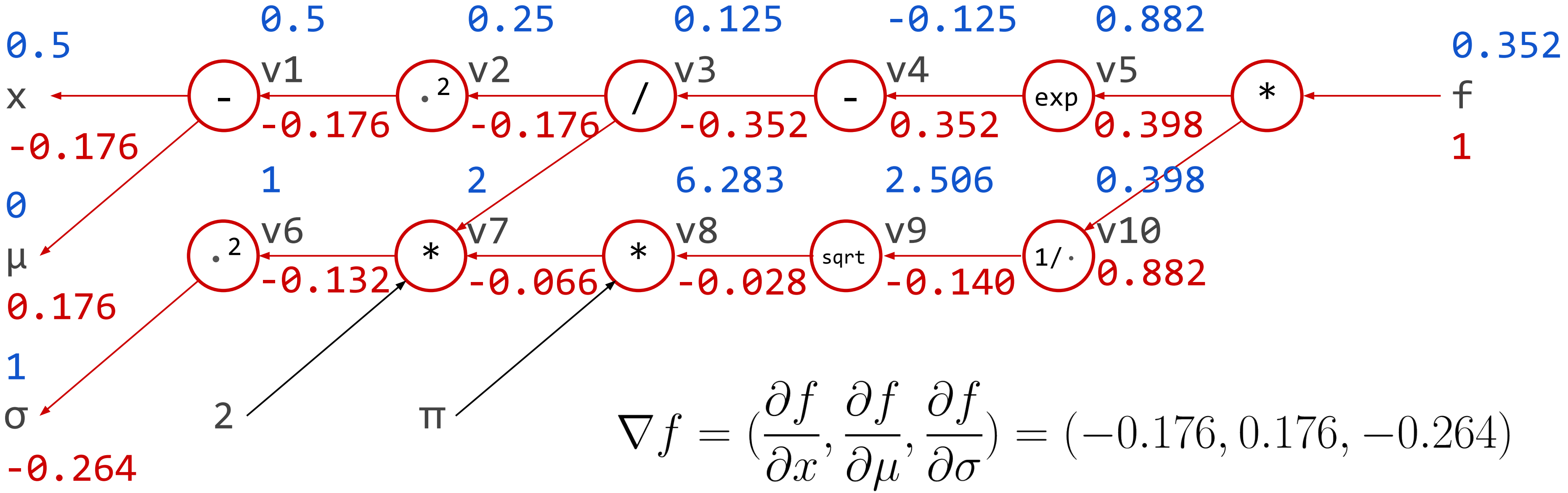


Normal PDF

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



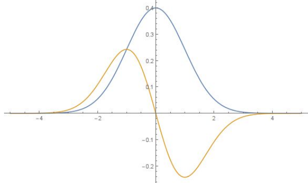
$$\frac{\partial f}{\partial x} = \frac{(\mu - x)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3} \qquad \frac{\partial f}{\partial \mu} = \frac{(x - \mu)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3} \qquad \frac{\partial f}{\partial \sigma} = -\frac{(\sigma - x + \mu)(\sigma + x - \mu)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^4}$$



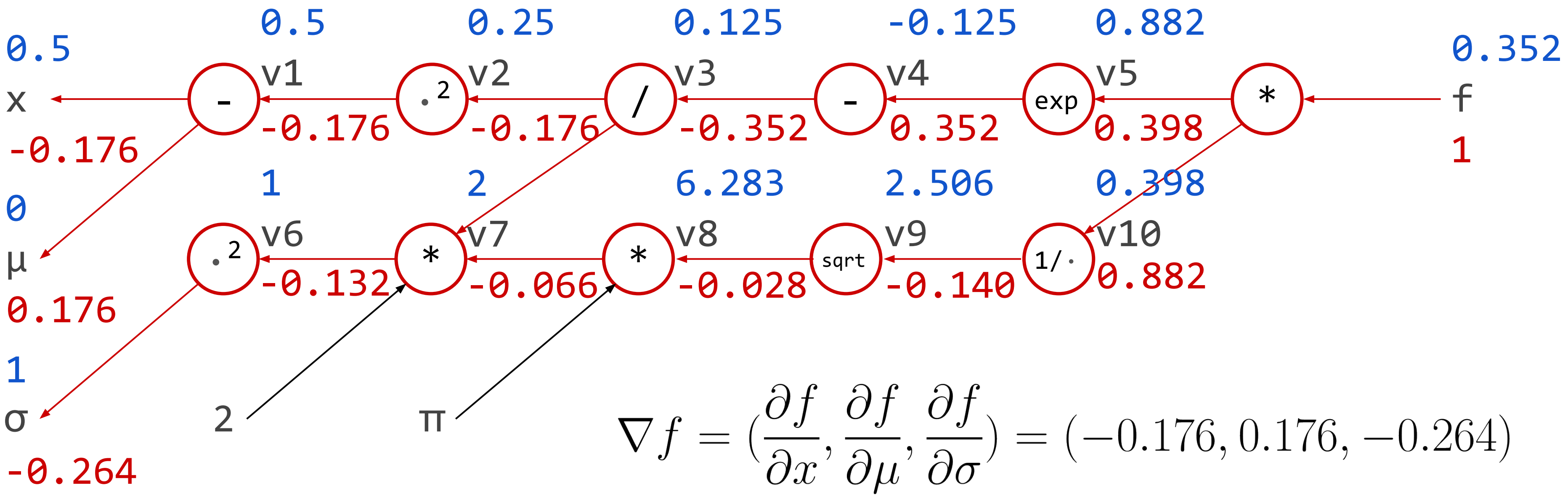
$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial \mu}, \frac{\partial f}{\partial \sigma}\right) = (-0.176, 0.176, -0.264)$$

Implementation

Where does the graph come from?

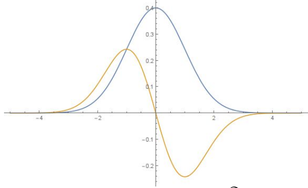


$$\frac{\partial f}{\partial x} = \frac{(\mu - x)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3} \qquad \frac{\partial f}{\partial \mu} = \frac{(x - \mu)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3} \qquad \frac{\partial f}{\partial \sigma} = -\frac{(\sigma - x + \mu)(\sigma + x - \mu)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^4}$$



$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial \mu}, \frac{\partial f}{\partial \sigma}\right) = (-0.176, 0.176, -0.264)$$

Where does the graph come from?



$$\frac{\partial f}{\partial x} = \frac{(\mu - x)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3} \qquad \frac{\partial f}{\partial \mu} = \frac{(x - \mu)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3} \qquad \frac{\partial f}{\partial \sigma} = -\frac{(\sigma - x + \mu)(\sigma + x - \mu)e^{-\frac{(\mu-x)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^4}$$

0.5

x

-0.176

0

μ

0.176

1

σ

-0.264

```
import torch, math
x = torch.tensor(0.5,requires_grad=True)
mu = torch.tensor(0., requires_grad=True)
sigma = torch.tensor(1., requires_grad=True)

p = (1/(torch.sqrt(2.*math.pi*sigma*sigma)))*torch.exp(-((x-mu)*(x-mu)/(2.*sigma*sigma)))
print(p)

p.backward()
print(x.grad, mu.grad, sigma.grad)
```

```
tensor(0.3521, grad_fn=<MulBackward0>)
tensor(-0.1760) tensor(0.1760) tensor(-0.2640)
```

2

π

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial \mu}, \frac{\partial f}{\partial \sigma}\right) = (-0.176, 0.176, -0.264)$$

Where does the graph come from?

Two main possibilities:

- **Static** computational graphs
  Let the user define the graph as a data structure

- **Dynamic** computational graphs
  Construct the graph automatically
  (general-purpose automatic differentiation)

Where does the graph come from?

Two main possibilities:

- **Static** computational graphs
  Let the user define the graph as a data structure

  "Define-and-run"

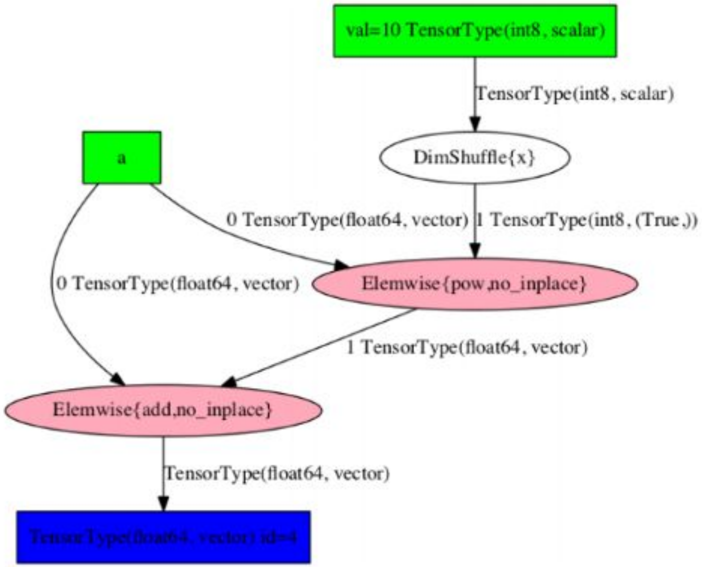
- **Dynamic** computational graphs
  Construct the graph automatically
  (general-purpose automatic differentiation)

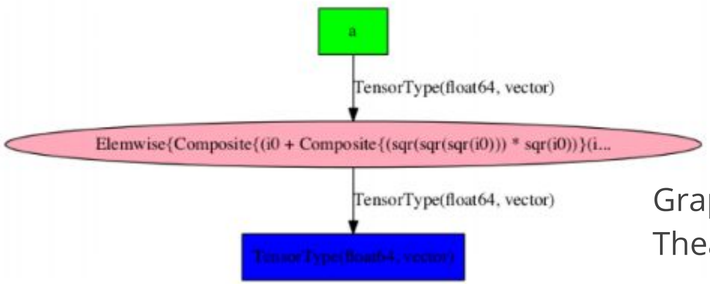
  "Define-by-run"

Static graphs (define-and-run)

Prototypical examples: Theano, TensorFlow

- The user creates the graph using **symbolic placeholders**, using a mini-language (domain-specific language, DSL)
- Limited (and unintuitive) control flow and expressivity
- The **graph gets “compiled”** to take care of expression swell, in-place ops.





Graph compilation in Theano

Static graphs (define-and-run)

Prototypical examples: Theano, TensorFlow

Let's implement A^k

Static graphs (define-and-run)

Prototypical examples: Theano, TensorFlow

Let's implement A^k

Pure Python:

```
result = 1
for i in range(k):
    result = result * A
```

Static graphs (define-and-run)

Prototypical examples: Theano, TensorFlow

Let's implement A^k

Pure Python:

```
result = 1
for i in range(k):
    result = result * A
```

theano

```
import theano
import theano.tensor as T

k = T.iscalar("k")
A = T.vector("A")

# Symbolic description of a loop
result, updates = theano.scan(fn=lambda prior_result, A: prior_result * A,
                              outputs_info=T.ones_like(A),
                              non_sequences=A,
                              n_steps=k)
final_result = result[-1]

# Compiled function that returns A**k
power = theano.function(inputs=[A,k], outputs=final_result, updates=updates)
```

Dynamic graphs (define-by-run)

Prototypical examples: PyTorch (and TensorFlow eager execution)

- General-purpose autodiff, usually via operator overloading
- The user **writes regular programs** in host programming language
  All language features (including control flow) are supported
- The **graph is automatically constructed**

Dynamic graphs (define-by-run)

Prototypical examples: PyTorch (and TensorFlow eager execution)

Let’s implement A^k

Dynamic graphs (define-by-run)

Prototypical examples: PyTorch (and TensorFlow eager execution)

Let's implement A^k

Pure Python:

```
result = 1
for i in range(k):
    result = result * A
```

Dynamic graphs (define-by-run)

Prototypical examples: PyTorch (and TensorFlow eager execution)

Let's implement A^k

Pure Python:

```
result = 1
for i in range(k):
    result = result * A
```

PyTorch

```
import torch

result = torch.tensor(1)
for i in range(k):
    result = result * A
```

```
result.backward()
print(A.grad)
```

Where to implement

Many possibilities

- Interpreter-based
- Compiler-based
  - Source code transformation
  - Operator overloading

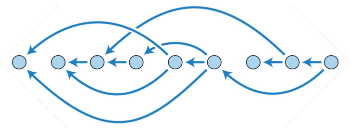
Where to implement

Many possibilities

- Interpreter-based
- Compiler-based
    - Source code transformation
    - **Operator overloading**

What to implement

Two main parts:



Computational graph

- **Dynamically build the graph**
  Side effect of forward evaluation or "non-standard interpretation"
- **Graph traversal algorithm**
  The API to kickstart the backpropagation:
  backward, grad, etc.

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

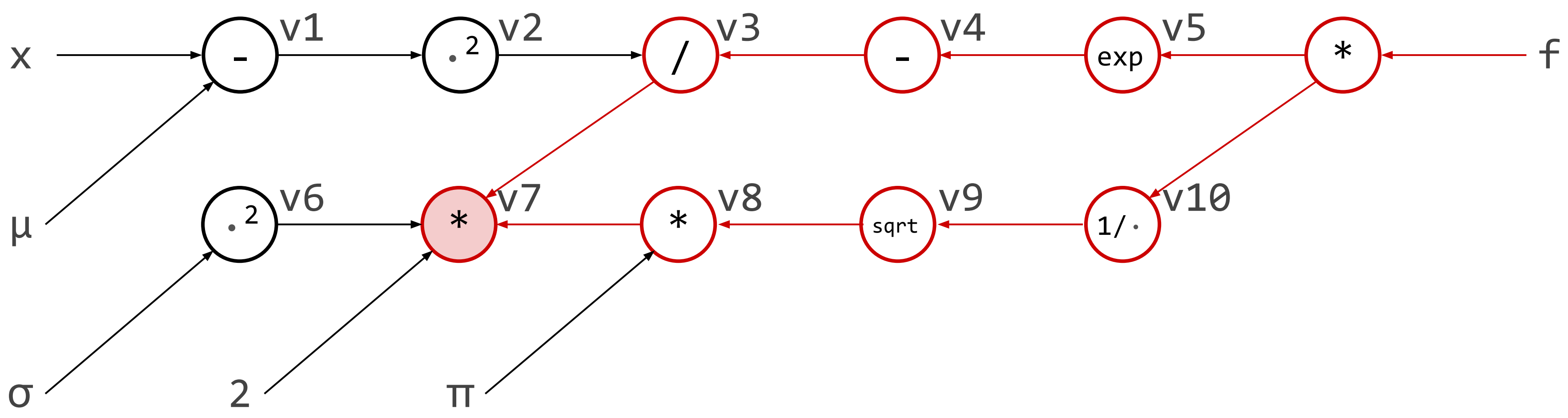
Derivatives

- **Rules of differentiation**
  For all elementary numerical operations, e.g., +, -, *, /, log, exp

  Usually implemented on a custom numerical type, using operator overloading

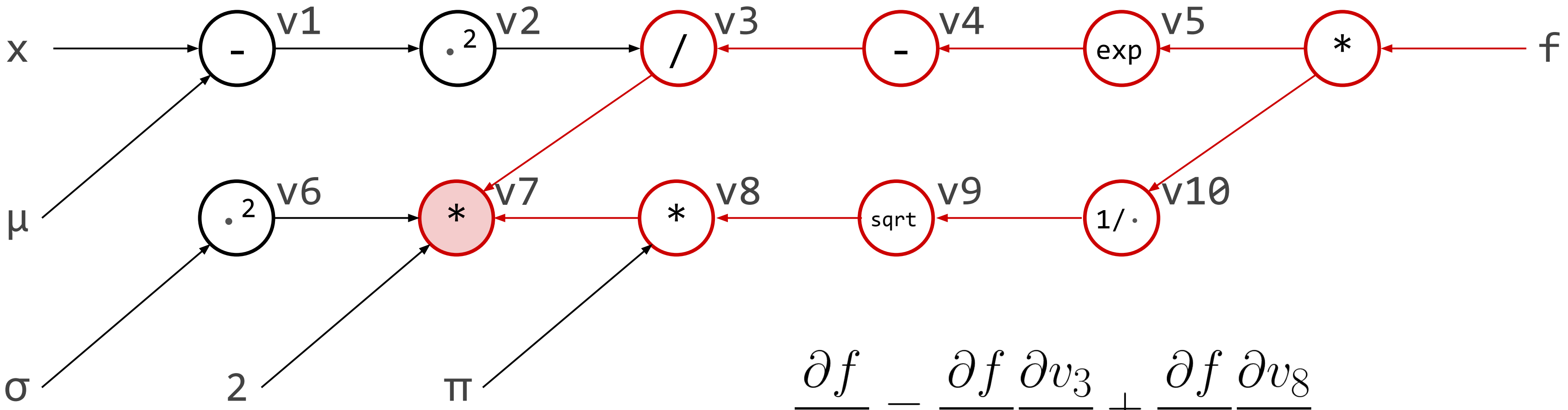
Correctly handle fan-out

Fan-out: when a node is involved in multiple subsequent operations



Correctly handle fan-out

Fan-out: when a node is involved in multiple subsequent operations

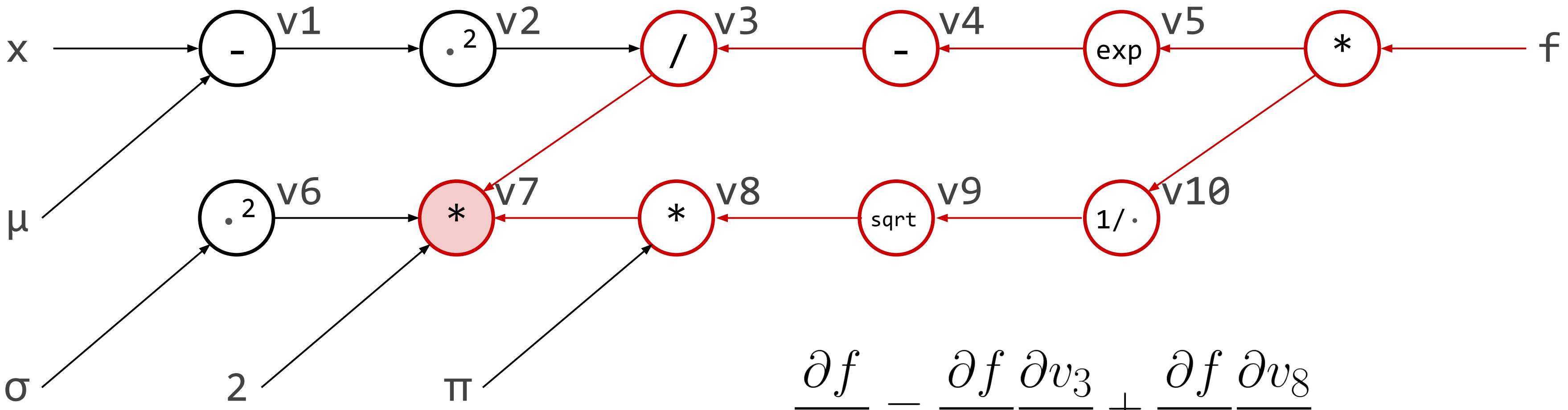


$$\frac{\partial f}{\partial v_7} = \frac{\partial f}{\partial v_3}\frac{\partial v_3}{\partial v_7} + \frac{\partial f}{\partial v_8}\frac{\partial v_8}{\partial v_7}$$

Correctly handle fan-out

Fan-out: when a node is involved in multiple subsequent operations

- Maintain a fan-out counter per node
- Don't propagate backward from a node until all derivatives coming to that node have arrived



$$\frac{\partial f}{\partial v_7} = \frac{\partial f}{\partial v_3}\frac{\partial v_3}{\partial v_7} + \frac{\partial f}{\partial v_8}\frac{\partial v_8}{\partial v_7}$$

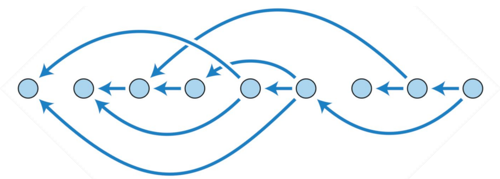
Operator overloading on custom type

```python
def eval_and_backprop(fun, x):
    y = fun(x)
    y.backprop(1.)
    return y, x._adjoint
```

```python
class OpMul():
    def __init__(a, b):
        self._a = a
        self._b = b
        return a * b

    def backprop(self, adjoint):
        self._a.backprop(adjoint * b)
        self._b.backprop(adjoint * a)
```

```python
class Number():
    def __init__(op, fan_out=0):
        self._op = op
        self._adjoint = 0.
        self._fan_out = fan_out

    def backprop(self, adjoint):
        self._adjoint += adjoint
        self._fan_out -= 1
        if self._fan_out == 0:
            self._op.backprop(self._adjoint)

    def __mul__(self, other):
        self._fan_out += 1
        other._fan_out += 1
        return Number(OpMul(self, other))
```

Graph with children pointing to parent(s)

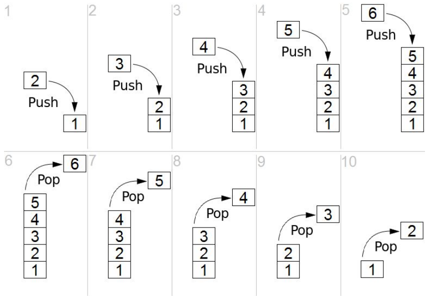


Operator overloading on custom type

```python
def eval_and_backprop(fun, x):
    y = fun(x)
    y.backprop(1.)
    return y, x._adjoint
```

```python
class OpMul():
    def __init__(a, b):
        self._a = a
        self._b = b
        return a * b

    def backprop(self, adjoint):
        self._a.backprop(adjoint * b)
        self._b.backprop(adjoint * a)
```

```python
tape = []

class Number():
    def __init__(value):
        self._value = value
        self._adjoint = 0.

    def backprop(self, adjoint):
        self._adjoint += adjoint

    def __mul__(self, other):
        global tape
        op = OpMul(self, other)
        tape.append(OpMul(self, other))
        return Number(self * other)
```



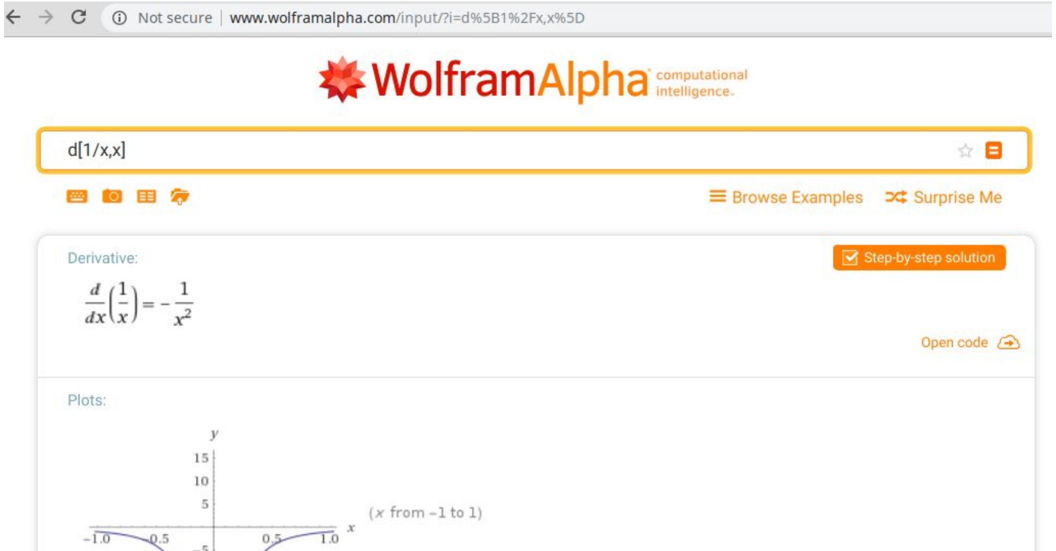
Global tape (stack) that records all ops.

- Forward: push in the order of evaluation
- Reverse: pop in the reverse order

Check for correctness

Use numerical and symbolic differentiation to check individual rules and your chain rule implementation

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h} + O(h^2)$$



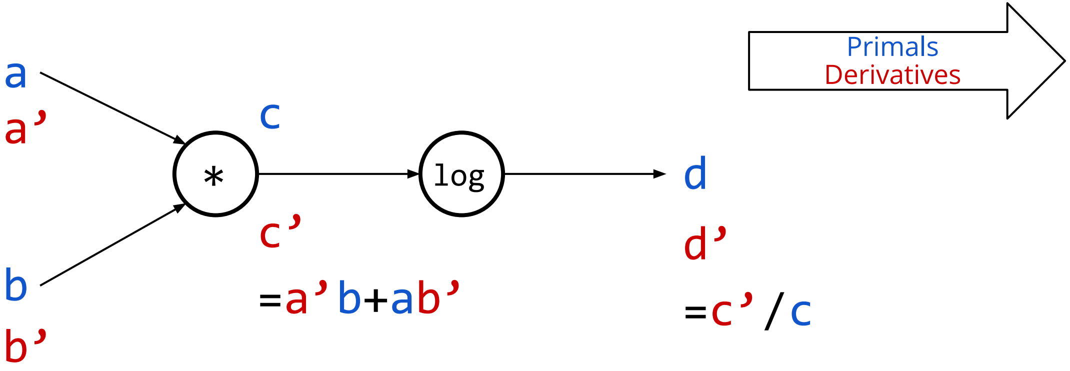
Advanced concepts

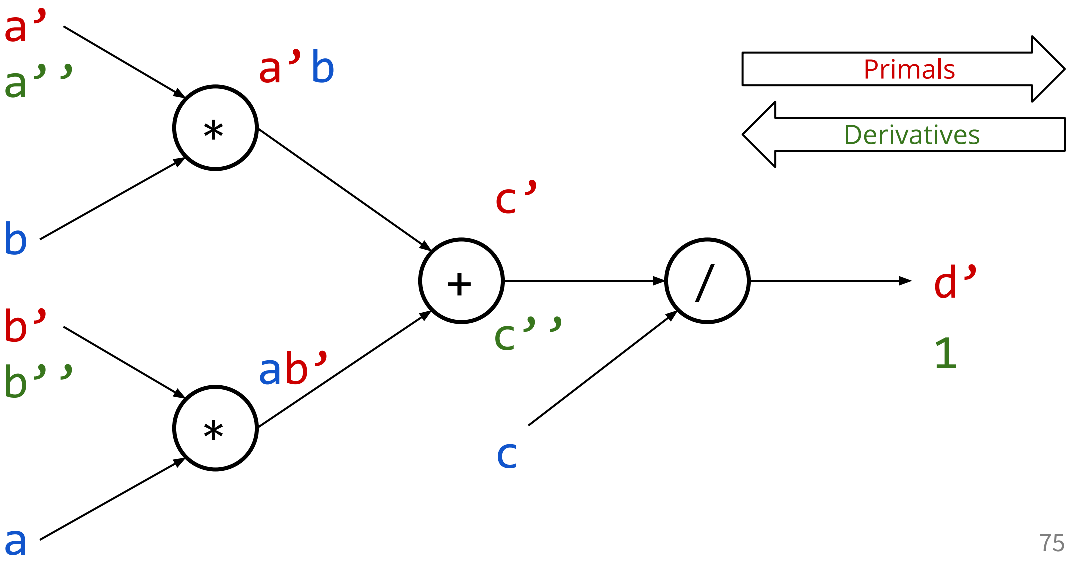
Nesting

Hessian–vector product (Pearlmutter, 1994) with reverse-on-forward

$$f : \mathbb{R}^2 \to \mathbb{R}$$

```
f(a, b):
   c = a * b
   d = log(c)
   return d

f(a, b)
```



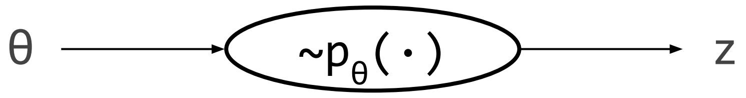


Random variables

Many modern machine learning models are probabilistic

- Need to optimize parameters of distributions from which we sample
- How do we take derivatives w.r.t. distribution parameters?

$$z \sim p_\theta(z)$$

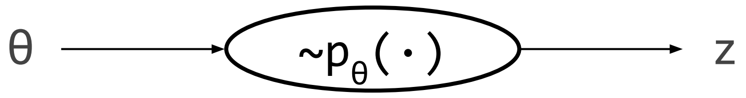


Random variables

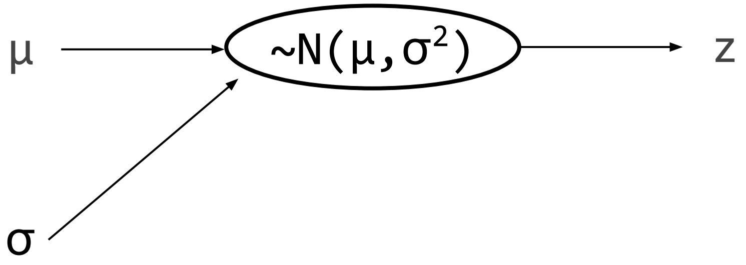
Many modern machine learning models are probabilistic

- Need to optimize parameters of distributions from which we sample
- How do we take derivatives w.r.t. distribution parameters?

$$z \sim p_\theta(z)$$



$$z \sim \mathcal{N}(\mu, \sigma^2)$$



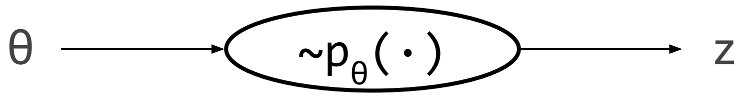
Random variables

Many modern machine learning models are probabilistic

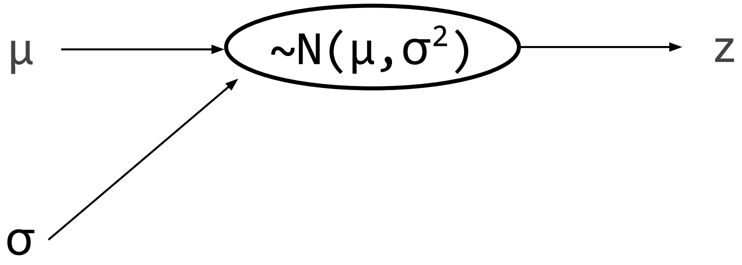
- Need to optimize parameters of distributions from which we sample
- How do we take derivatives w.r.t. distribution parameters?

Problem: sampling **is not a differentiable operation**

$$z \sim p_\theta(z)$$

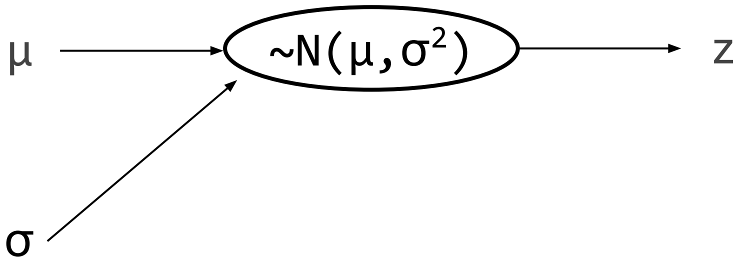


$$z \sim \mathcal{N}(\mu, \sigma^2)$$



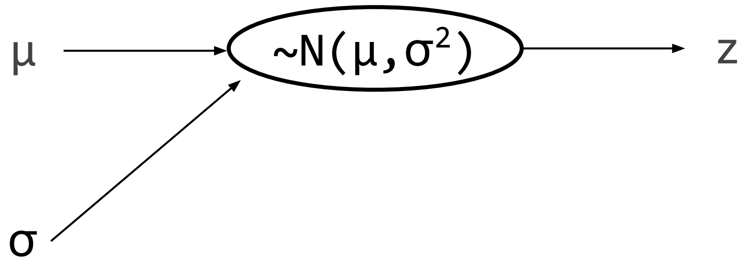
Random variables

$$z \sim \mathcal{N}(\mu, \sigma^2)$$



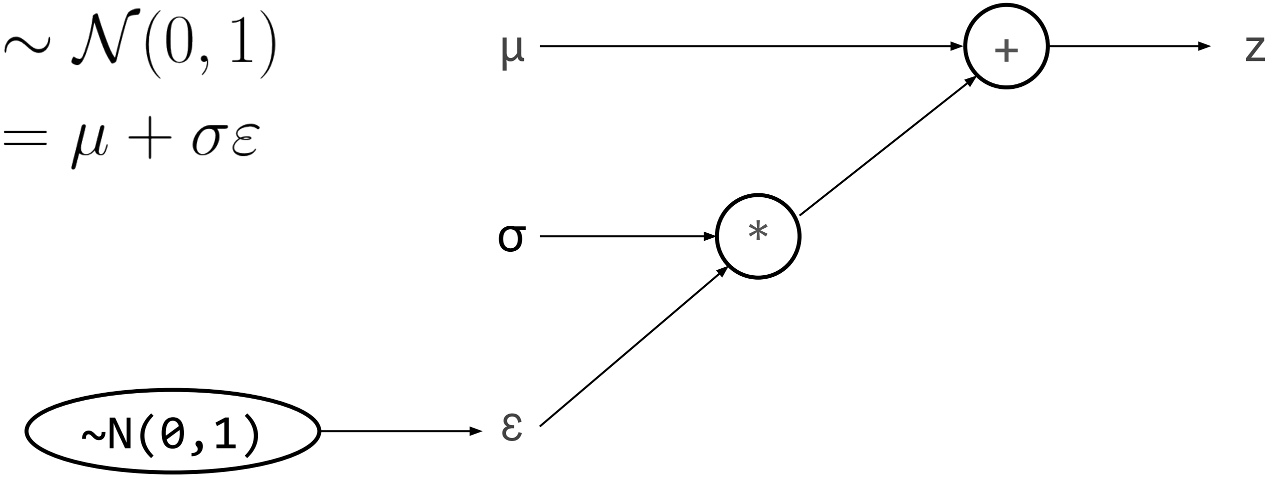
Random variables

$$z \sim \mathcal{N}(\mu, \sigma^2)$$



$$\varepsilon \sim \mathcal{N}(0, 1)$$

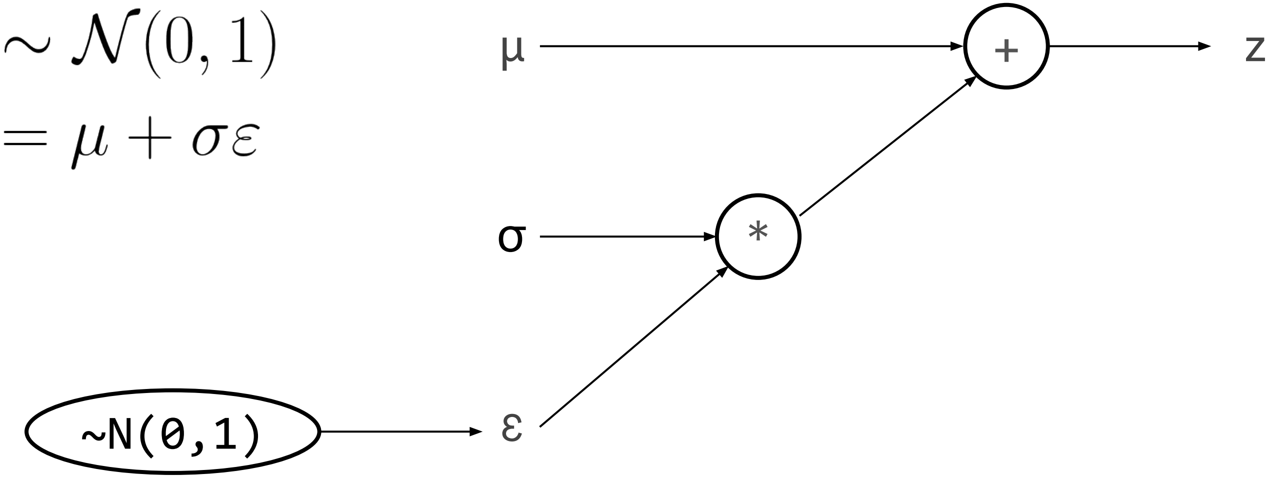
$$z = \mu + \sigma\varepsilon$$



Random variables

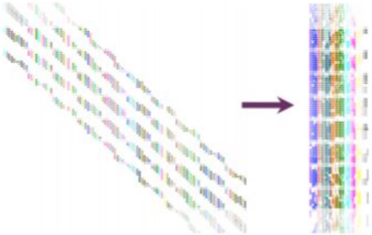
- Somewhat obvious in a general-purpose autodiff code
- Known as "reparameterization trick" used by variational autoencoders (VAEs) (Kingma & Welling, 2014); Stochastic backpropagation (Rezende et al., 2014)

$$\varepsilon \sim \mathcal{N}(0,1)$$
$$z = \mu + \sigma\varepsilon$$

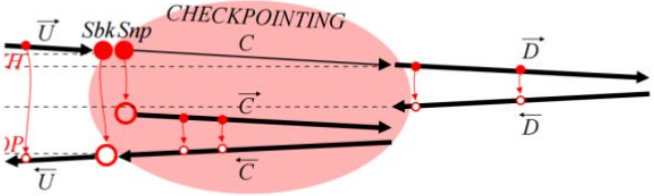


Other advanced concepts

- Tape reduction and elimination (Naumann, 2004)
- Context-aware source-to-source transformation (Utke, 2004)
- Sparsity-aware autodiff by matrix coloring (Gebremedhin et al., 2013)



- Reverse mode checkpointing (Dauvergne & Hascoet, 2006)



https://github.com/openai/gradient-checkpointing

Gruslys, A., Munos, R., Danihelka, I., Lanctot, M. and Graves, A., 2016. Memory-efficient backpropagation through time. NIPS 2016

Summary

Summary

- Derivatives in machine learning
- How do we compute derivatives: manual, symbolic, numerical, autodiff
- Automatic differentiation
  - Forward and reverse: $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ use reverse when $n \gg m$
  - Keep forward in mind if you need something more than backprop
- Computational graphs and propagation
- Implementation
  - Where does the graph come from?
  - Strategies and performance tips
- Advanced concepts
  - Reparameterization
  - Nesting, higher-order derivatives,checkpointing

References

Baydin, A.G., Pearlmutter, B.A., Radul, A.A. and Siskind, J.M., 2017. Automatic differentiation in machine learning: a survey. Journal of Machine Learning Research (JMLR), 18(153), pp.1-153.

Baydin, Atılım Güneş, Barak A. Pearlmutter, and Jeffrey Mark Siskind. 2016. "Tricks from Deep Learning." In 7th International Conference on Algorithmic Differentiation, Christ Church Oxford, UK, September 12–15, 2016.

Baydin, Atılım Güneş, Barak A. Pearlmutter, and Jeffrey Mark Siskind. 2016. "DiffSharp: An AD Library for .NET Languages." In 7th International Conference on Algorithmic Differentiation, Christ Church Oxford, UK, September 12–15, 2016.

Baydin, Atılım Güneş, Robert Cornish, David Martínez Rubio, Mark Schmidt, and Frank Wood. 2018. "Online Learning Rate Adaptation with Hypergradient Descent." In Sixth International Conference on Learning Representations (ICLR), Vancouver, Canada, April 30 – May 3, 2018.

Griewank, A. and Walther, A., 2008. Evaluating derivatives: principles and techniques of algorithmic differentiation (Vol. 105). SIAM.

Nocedal, J. and Wright, S.J., 1999. Numerical Optimization. Springer.

Extra slides

Forward vs reverse summary

In the extreme $\mathbf{f} : \mathbb{R} \to \mathbb{R}^m$
use forward mode to evaluate

$$(\frac{\partial f_1}{\partial x}, \cdots, \frac{\partial f_m}{\partial x})$$

In the extreme $f : \mathbb{R}^n \to \mathbb{R}$
use reverse mode to evaluate

$$\nabla f(\mathbf{x}) = (\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n})$$

In general $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ the Jacobian $\mathbf{J}_f(\mathbf{x}) \in \mathbb{R}^{m \times n}$ can be evaluated in

- $O(n\,\mathrm{time}(\mathbf{f}))$ with forward mode
- $O(m\,\mathrm{time}(\mathbf{f}))$ with reverse mode

Reverse performs better when $n \gg m$

Current Landscape

Currently in progress: frameworks are in transition from

coarse-grained (module level) backprop

towards

fine-grained, general-purpose automatic differentiation

torch7 *2011* → **torch-autograd** *2015* → **PyTorch** *2016*

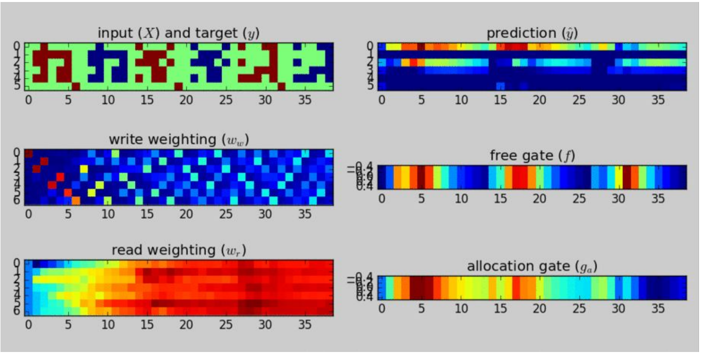
HIPS autograd *2014* → **torch-autograd** *2015*

TensorFlow *2015* → **TensorFlow eager execution** *2017*

Current Landscape

A new mindset and workflow, enabling differentiable algorithmic elements

- Neural Turing Machine, Differentiable Neural Computer (Graves et al. 2014, 2016)
    - Can infer algorithms: copy, sort, recall
- Stack-augmented RNN (Joulin & Mikolov, 2015)
- End-to-end memory network (Sukhbaatar et al., 2015)
- Stack, queue, deque (Grefenstette et al., 2015)
- Discrete interfaces (Zaremba & Sutskever, 2015)



DNC on binary number recall
(Wikimedia Commons: Kjerish)

Current Landscape

General-purpose AD enables new libraries such as Pyro

Example:

Pyro supports stochastic recursion, higher-order functions, random control flow and runs stochastic variational inference enabled by PyTorch autograd infrastructure

PyTorch *2016* → **Pyro** *2017*

```
In [7]: def geometric(p, t=None):
            if t is None:
                t = 0
            x = pyro.sample("x_{}".format(t), dist.bernoulli, p)
            if torch.equal(x.data, torch.zeros(1)):
                return x
            else:
                return x + geometric(p, t+1)

        print(geometric(Variable(torch.Tensor([0.5]))))

        Variable containing:
         0
        [torch.FloatTensor of size 1]
```

Dynamically generating random variables

DiffSharp

	Op.	Value	Type signature	AD	Num.	Sym.
$f : \mathbb{R} \to \mathbb{R}$	`diff`	f'	$(\mathbb{R} \to \mathbb{R}) \to \mathbb{R} \to \mathbb{R}$	X, F	A	X
	`diff'`	(f, f')	$(\mathbb{R} \to \mathbb{R}) \to \mathbb{R} \to (\mathbb{R} \times \mathbb{R})$	X, F	A	X
	`diff2`	f''	$(\mathbb{R} \to \mathbb{R}) \to \mathbb{R} \to \mathbb{R}$	X, F	A	X
	`diff2'`	(f, f'')	$(\mathbb{R} \to \mathbb{R}) \to \mathbb{R} \to (\mathbb{R} \times \mathbb{R})$	X, F	A	X
	`diff2''`	(f, f', f'')	$(\mathbb{R} \to \mathbb{R}) \to \mathbb{R} \to (\mathbb{R} \times \mathbb{R} \times \mathbb{R})$	X, F	A	X
	`diffn`	$f^{(n)}$	$\mathbb{N} \to (\mathbb{R} \to \mathbb{R}) \to \mathbb{R} \to \mathbb{R}$	X, F		X
	`diffn'`	$(f, f^{(n)})$	$\mathbb{N} \to (\mathbb{R} \to \mathbb{R}) \to \mathbb{R} \to (\mathbb{R} \times \mathbb{R})$	X, F		X
$f : \mathbb{R}^n \to \mathbb{R}$	`grad`	∇f	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to \mathbb{R}^n$	X, R	A	X
	`grad'`	$(f, \nabla f)$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to (\mathbb{R} \times \mathbb{R}^n)$	X, R	A	X
	`gradv`	$\nabla f \cdot \mathbf{v}$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to \mathbb{R}^n \to \mathbb{R}$	X, F	A	
	`gradv'`	$(f, \nabla f \cdot \mathbf{v})$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to \mathbb{R}^n \to (\mathbb{R} \times \mathbb{R})$	X, F	A	
	`hessian`	$\mathbf{H}_f$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to \mathbb{R}^{n \times n}$	X, R-F	A	X
	`hessian'`	$(f, \mathbf{H}_f)$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to (\mathbb{R} \times \mathbb{R}^{n \times n})$	X, R-F	A	X
	`hessianv`	$\mathbf{H}_f \mathbf{v}$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to \mathbb{R}^n \to \mathbb{R}^n$	X, F-R	A	
	`hessianv'`	$(f, \mathbf{H}_f \mathbf{v})$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to \mathbb{R}^n \to (\mathbb{R} \times \mathbb{R}^n)$	X, F-R	A	
	`gradhessian`	$(\nabla f, \mathbf{H}_f)$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to (\mathbb{R}^n \times \mathbb{R}^{n \times n})$	X, R-F	A	X
	`gradhessian'`	$(f, \nabla f, \mathbf{H}_f)$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to (\mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^{n \times n})$	X, R-F	A	X
	`gradhessianv`	$(\nabla f \cdot \mathbf{v}, \mathbf{H}_f \mathbf{v})$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to \mathbb{R}^n \to (\mathbb{R} \times \mathbb{R}^n)$	X, F-R	A	
	`gradhessianv'`	$(f, \nabla f \cdot \mathbf{v}, \mathbf{H}_f \mathbf{v})$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to \mathbb{R}^n \to (\mathbb{R} \times \mathbb{R} \times \mathbb{R}^n)$	X, F-R	A	
	`laplacian`	$\mathrm{tr}(\mathbf{H}_f)$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to \mathbb{R}$	X, R-F	A	X
	`laplacian'`	$(f, \mathrm{tr}(\mathbf{H}_f))$	$(\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^n \to (\mathbb{R} \times \mathbb{R})$	X, R-F	A	X
$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$	`jacobian`	$\mathbf{J}_\mathbf{f}$	$(\mathbb{R}^n \to \mathbb{R}^m) \to \mathbb{R}^n \to \mathbb{R}^{m \times n}$	X, F/R	A	X
	`jacobian'`	$(\mathbf{f}, \mathbf{J}_\mathbf{f})$	$(\mathbb{R}^n \to \mathbb{R}^m) \to \mathbb{R}^n \to (\mathbb{R}^m \times \mathbb{R}^{m \times n})$	X, F/R	A	X
	`jacobianv`	$\mathbf{J}_\mathbf{f} \mathbf{v}$	$(\mathbb{R}^n \to \mathbb{R}^m) \to \mathbb{R}^n \to \mathbb{R}^n \to \mathbb{R}^m$	X, F	A	
	`jacobianv'`	$(\mathbf{f}, \mathbf{J}_\mathbf{f} \mathbf{v})$	$(\mathbb{R}^n \to \mathbb{R}^m) \to \mathbb{R}^n \to \mathbb{R}^n \to (\mathbb{R}^m \times \mathbb{R}^m)$	X, F	A	
	`jacobianT`	$\mathbf{J}_\mathbf{f}^T$	$(\mathbb{R}^n \to \mathbb{R}^m) \to \mathbb{R}^n \to \mathbb{R}^{n \times m}$	X, F/R	A	X
	`jacobianT'`	$(\mathbf{f}, \mathbf{J}_\mathbf{f}^T)$	$(\mathbb{R}^n \to \mathbb{R}^m) \to \mathbb{R}^n \to (\mathbb{R}^m \times \mathbb{R}^{n \times m})$	X, F/R	A	X
	`jacobianTv`	$\mathbf{J}_\mathbf{f}^T \mathbf{v}$	$(\mathbb{R}^n \to \mathbb{R}^m) \to \mathbb{R}^n \to \mathbb{R}^m \to \mathbb{R}^n$	X, R		
	`jacobianTv'`	$(\mathbf{f}, \mathbf{J}_\mathbf{f}^T \mathbf{v})$	$(\mathbb{R}^n \to \mathbb{R}^m) \to \mathbb{R}^n \to \mathbb{R}^m \to (\mathbb{R}^m \times \mathbb{R}^n)$	X, R		
	`jacobianTv''`	$(\mathbf{f}, \mathbf{J}_\mathbf{f}^T(\cdot))$	$(\mathbb{R}^n \to \mathbb{R}^m) \to \mathbb{R}^n \to (\mathbb{R}^m \times (\mathbb{R}^m \to \mathbb{R}^n))$	X, R		
	`curl`	$\nabla \times \mathbf{f}$	$(\mathbb{R}^3 \to \mathbb{R}^3) \to \mathbb{R}^3 \to \mathbb{R}^3$	X, F	A	X
	`curl'`	$(\mathbf{f}, \nabla \times \mathbf{f})$	$(\mathbb{R}^3 \to \mathbb{R}^3) \to \mathbb{R}^3 \to (\mathbb{R}^3 \times \mathbb{R}^3)$	X, F	A	X
	`div`	$\nabla \cdot \mathbf{f}$	$(\mathbb{R}^n \to \mathbb{R}^n) \to \mathbb{R}^n \to \mathbb{R}$	X, F	A	X
	`div'`	$(\mathbf{f}, \nabla \cdot \mathbf{f})$	$(\mathbb{R}^n \to \mathbb{R}^n) \to \mathbb{R}^n \to (\mathbb{R}^n \times \mathbb{R})$	X, F	A	X
	`curldiv`	$(\nabla \times \mathbf{f}, \nabla \cdot \mathbf{f})$	$(\mathbb{R}^3 \to \mathbb{R}^3) \to \mathbb{R}^3 \to (\mathbb{R}^3 \times \mathbb{R})$	X, F	A	X
	`curldiv'`	$(\mathbf{f}, \nabla \times \mathbf{f}, \nabla \cdot \mathbf{f})$	$(\mathbb{R}^3 \to \mathbb{R}^3) \to \mathbb{R}^3 \to (\mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R})$	X, F	A	X