PyTorch

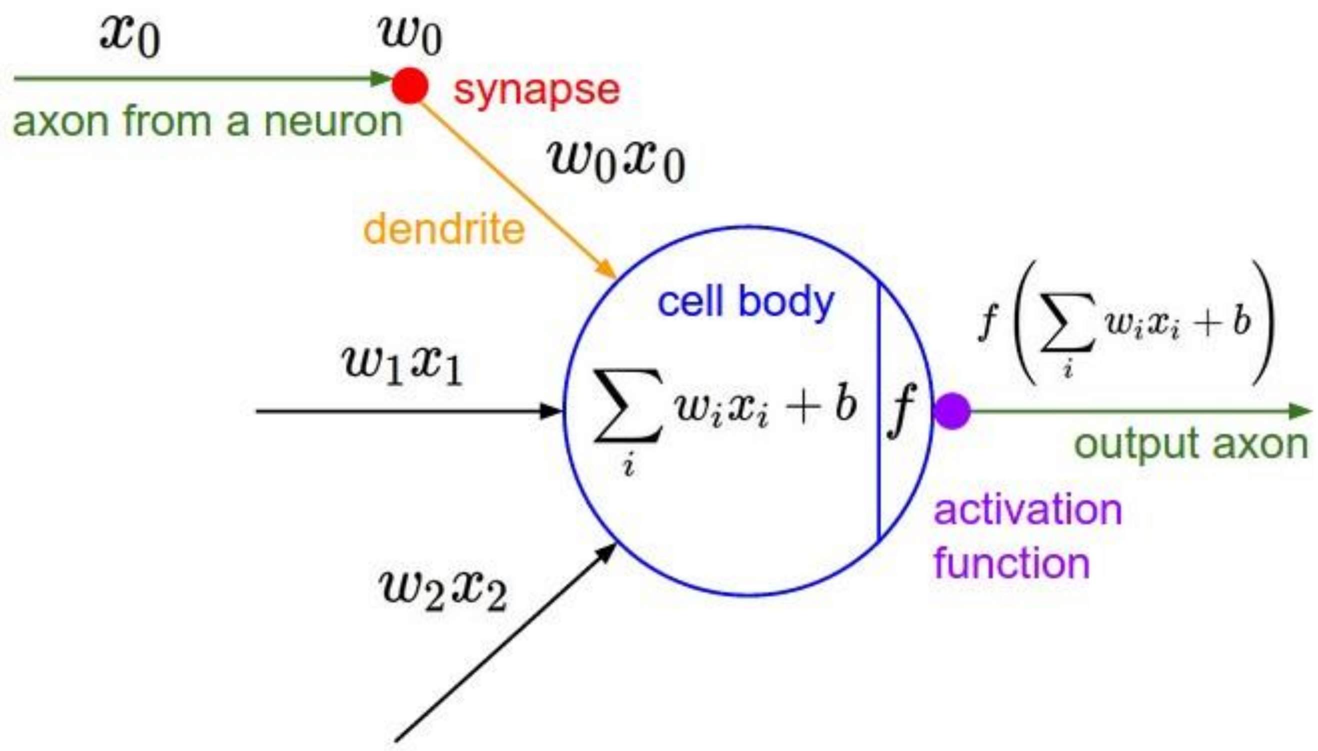
激活函数及其梯度

主讲人：龙良曲

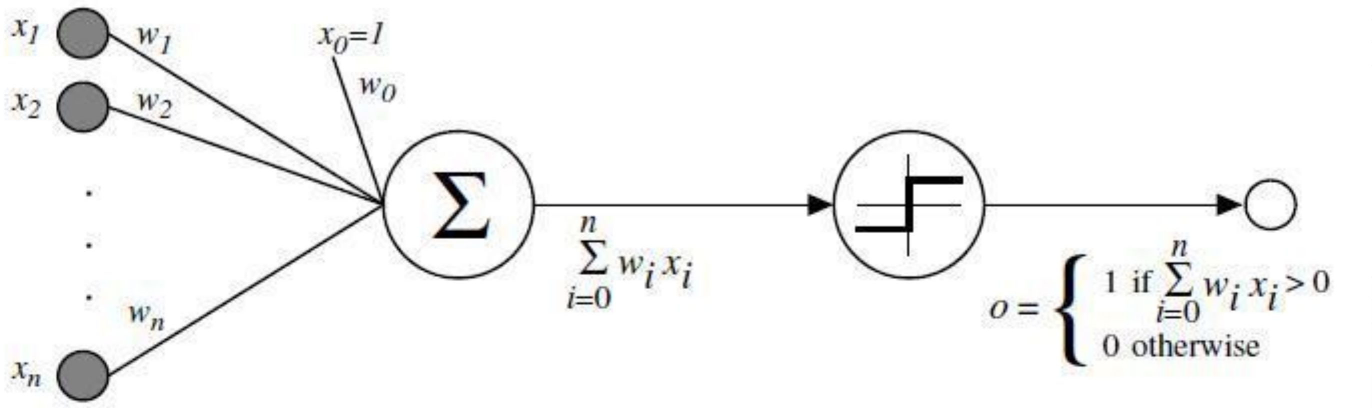
Activation Functions

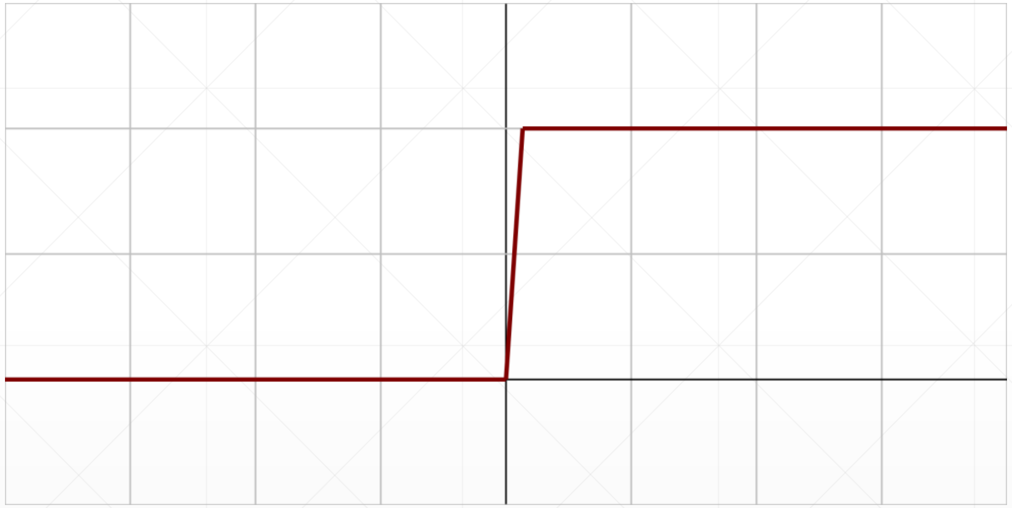
PITTS WITH LETTVIN: Pitts with Jerome Lettvin and one subject of their experiments on visual perception (1959).

Wikipedia



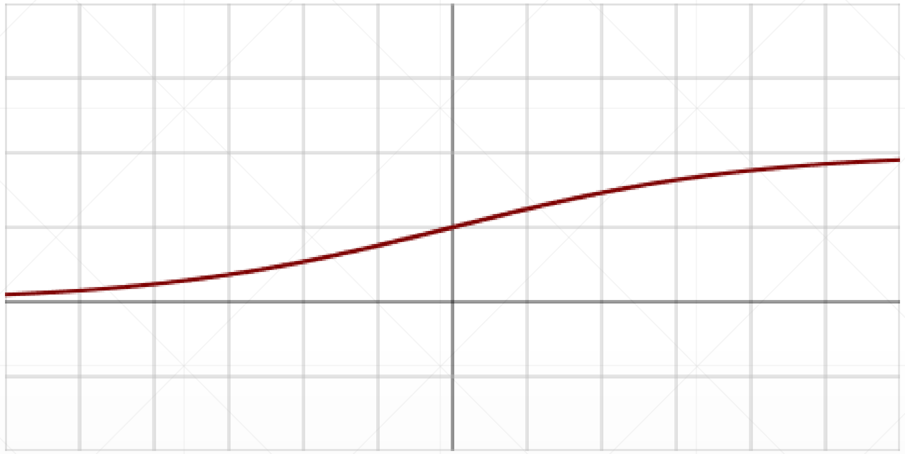
Derivative


x_1
w_1
x_2
w_2
$x_0=1$
w_0
w_n
x_n
Σ
$\sum_{i=0}^{n} w_i x_i$
$o = \begin{cases} 1 \text{ if } \sum_{i=0}^{n} w_i x_i > 0 \\ 0 \text{ otherwise} \end{cases}$



Sigmoid / Logistic

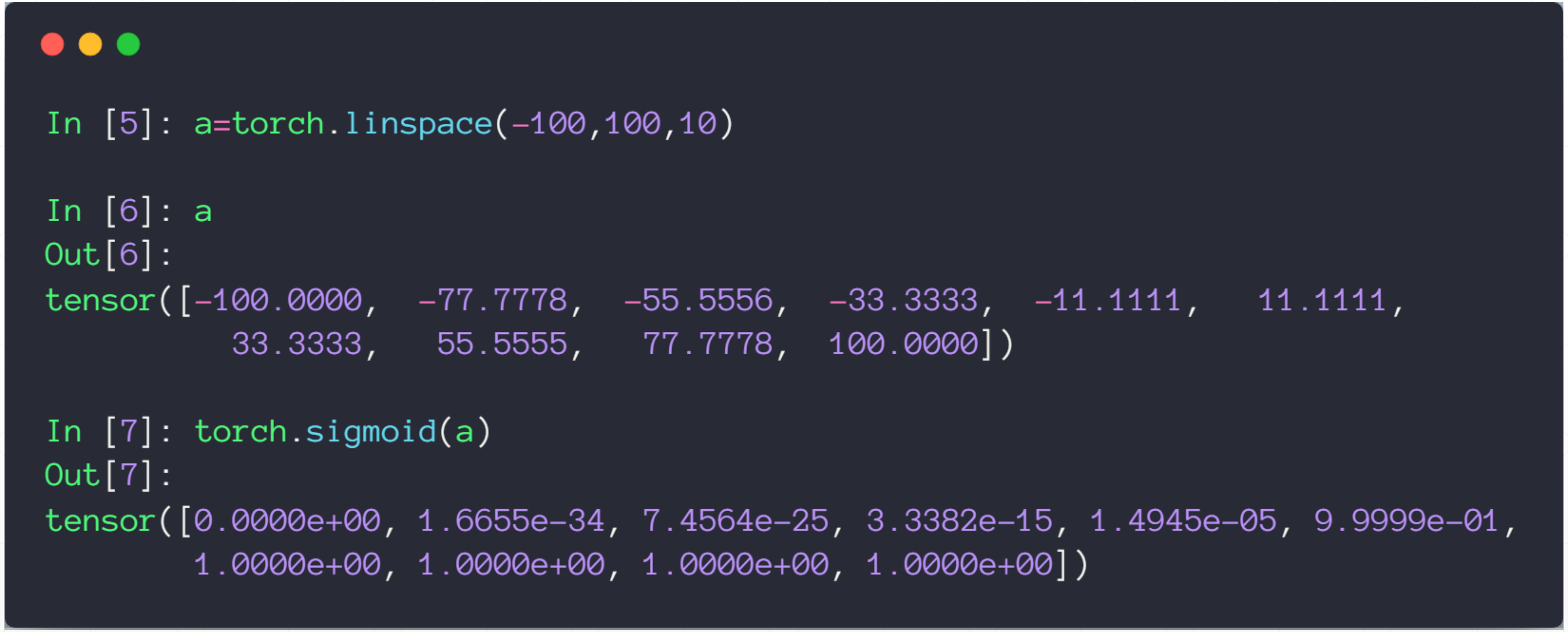
$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$



Derivative

$$
\begin{aligned}
\frac{d}{dx}\sigma(x) &= \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{(1+e^{-x})-1}{(1+e^{-x})^2} \\
&= \frac{1+e^{-x}}{(1+e^{-x})^2} - \left(\frac{1}{1+e^{-x}}\right)^2 \\
&= \sigma(x) - \sigma(x)^2 \\
\sigma' &= \sigma(1-\sigma)
\end{aligned}
$$

torch.sigmoid

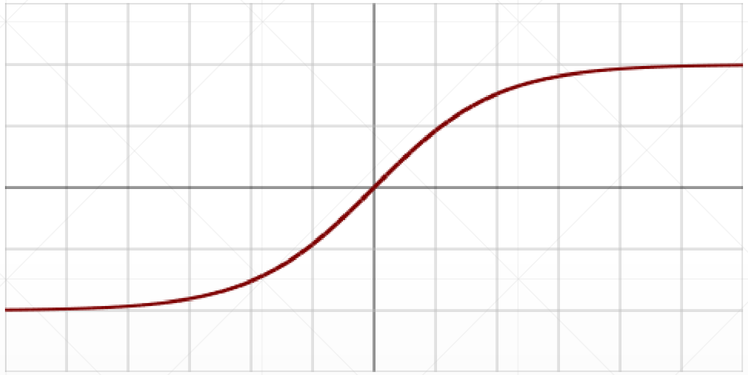


```
In [5]: a=torch.linspace(-100,100,10)

In [6]: a
Out[6]:
tensor([-100.0000,  -77.7778,  -55.5556,  -33.3333,  -11.1111,   11.1111,
          33.3333,   55.5555,   77.7778,  100.0000])

In [7]: torch.sigmoid(a)
Out[7]:
tensor([0.0000e+00, 1.6655e-34, 7.4564e-25, 3.3382e-15, 1.4945e-05, 9.9999e-01,
        1.0000e+00, 1.0000e+00, 1.0000e+00, 1.0000e+00])
```

Tanh

$$f(x) = \tanh(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$$

$$= 2\mathit{sigmoid}(2x) - 1$$

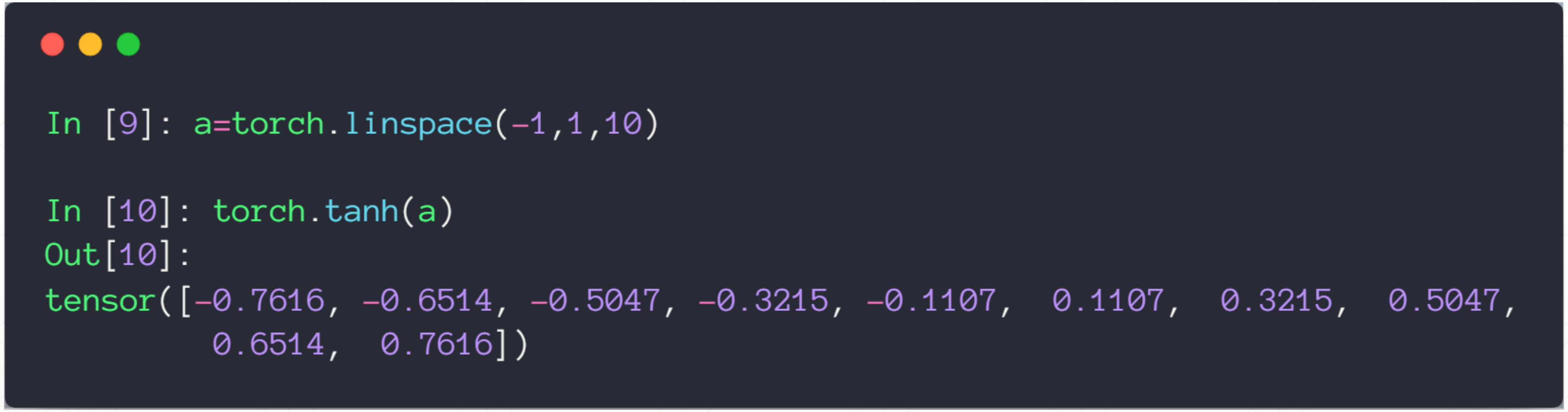


Derivative

$$\frac{d}{dx}\tanh(x) = \frac{(e^x + e^{-x})(e^x + e^{-x}) - (e^x - e^{-x})(e^x - e^{-x})}{(e^x + e^{-x})^2}$$

$$= 1 - \frac{(e^x - e^{-x})^2}{(e^x + e^{-x})^2} = 1 - \tanh^2(x)$$

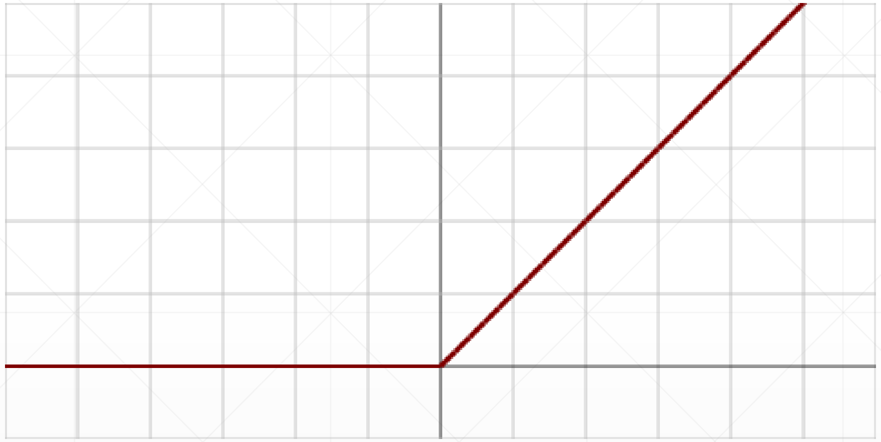
torch.tanh



```
In [9]: a=torch.linspace(-1,1,10)

In [10]: torch.tanh(a)
Out[10]:
tensor([-0.7616, -0.6514, -0.5047, -0.3215, -0.1107,  0.1107,  0.3215,  0.5047,
         0.6514,  0.7616])
```

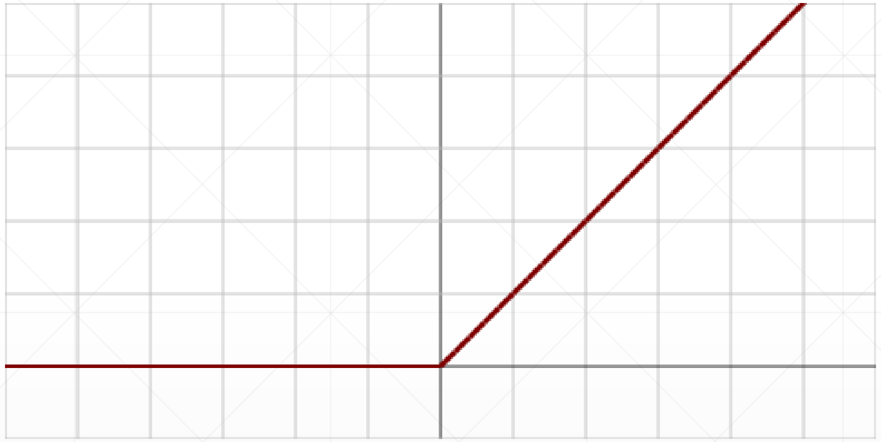
Rectified Linear Unit

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

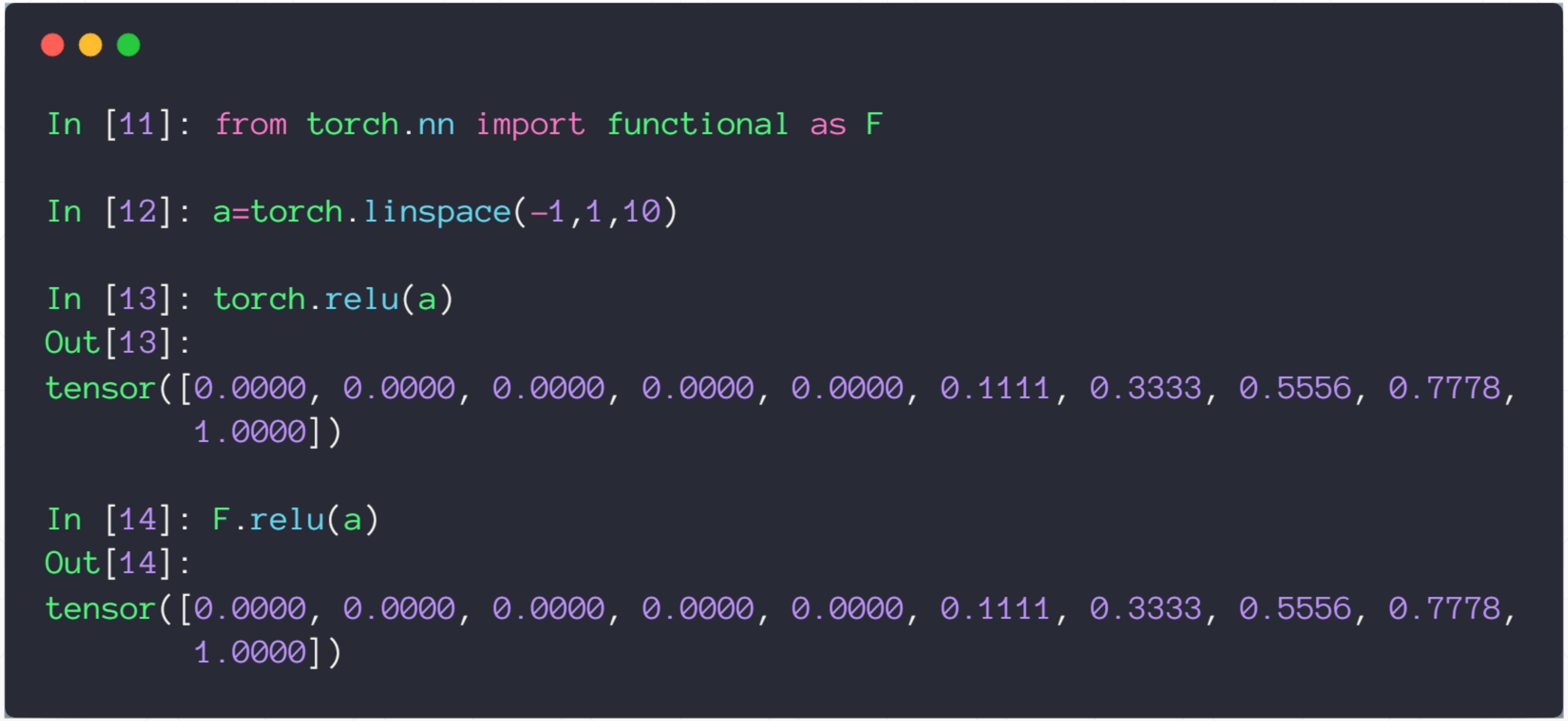


Derivative

$$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$



F.relu



```
In [11]: from torch.nn import functional as F

In [12]: a=torch.linspace(-1,1,10)

In [13]: torch.relu(a)
Out[13]:
tensor([0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.1111, 0.3333, 0.5556, 0.7778,
        1.0000])

In [14]: F.relu(a)
Out[14]:
tensor([0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.1111, 0.3333, 0.5556, 0.7778,
        1.0000])
```

下一课时

Loss及其梯度

Thank You.