

```

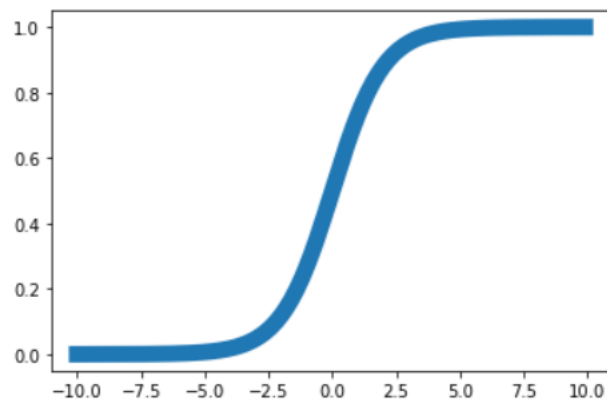
In [1]: %matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import math
#显示中文
plt.rcParams['font.sans-serif'] = ['Arial Unicode MS']

In [3]: def sigmoid(x):
        return 1/(1+np.exp(-x))

In [4]: def df_sigmoid(x):
        s = sigmoid(x)
        return np.multiply(s, (np.ones(len(x)) - s))

In [5]: x = np.arange(-10., 10., 0.1)
y_sigmoid = sigmoid(x)
#df = np.multiply(y_sigmoid, (np.ones(len(x)) - y_sigmoid))
df = df_sigmoid(x)
plt.plot(x, y_sigmoid, label = u'Sigmoid', linewidth=10)
plt.show()

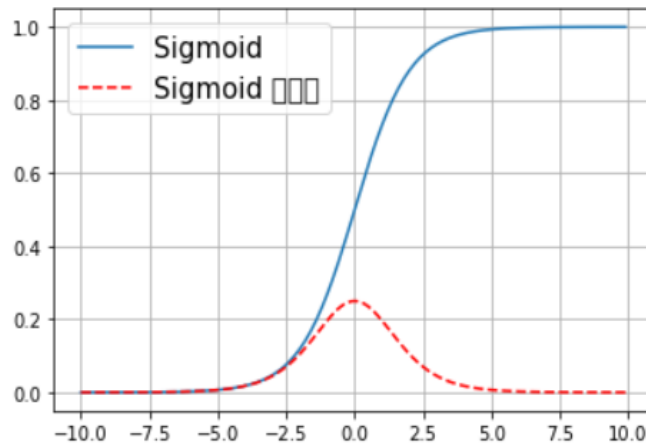
```



```

In [6]: x = np.arange(-10., 10., 0.1)
y_sigmoid = sigmoid(x)
#df = np.multiply(y_sigmoid, (np.ones(len(x)) - y_sigmoid))
df = df_sigmoid(x)
plt.plot(x, y_sigmoid, label = u'Sigmoid')
plt.plot(x, df, 'r', linestyle = '--', label = u'Sigmoid 的导数')
plt.grid() #生成网格线
plt.legend(fontsize = 15)

```

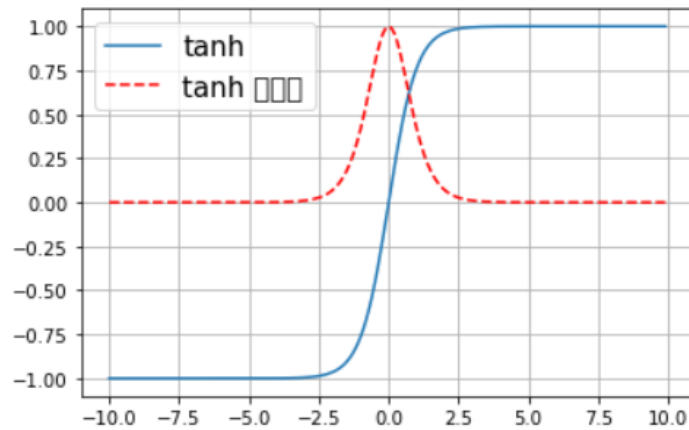


```
In [7]: def tanh(x):
        return np.tanh(x)
        #return (1-np.exp(-2x))/(1+np.exp(-2x))
```

```
In [8]: def df_tanh(x):
        t = tanh(x)
        return 1 - np.power(t, 2) #np.power() 幂次方
```

```
In [9]: x = np.arange(-10., 10., 0.1)
        y_tanh = tanh(x)
        df = df_tanh(x)
        plt.plot(x, y_tanh, label = u'tanh')
        plt.plot(x, df, 'r', linestyle = '--', label = u'tanh 的导数')
        plt.grid()
        plt.legend(fontsize = 15)
```

Out[9]: <matplotlib.legend.Legend at 0x1678f02cfd0>

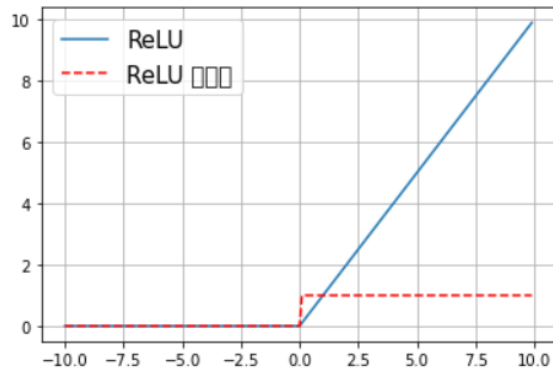


```
In [10]: def ReLU(x):
          arr = []
          for i in x:
              arr.append(0 if i<0 else i )
          return arr
```

```
In [11]: def df_ReLU(x):
          arr = []
          for i in x:
              arr.append(0 if i<0 else 1 )
          return arr
```

```
In [12]: x = np.arange(-10., 10., 0.1)
          y_ReLU = ReLU(x)
          df = df_ReLU(x)
          plt.plot(x, y_ReLU, label = u'ReLU')
          plt.plot(x, df, 'r', linestyle = '--', label = u'ReLU 的导数')
          plt.grid()
          plt.legend(fontsize = 15)
```

```
Out[12]: <matplotlib.legend.Legend at 0x1678eefedc0>
```

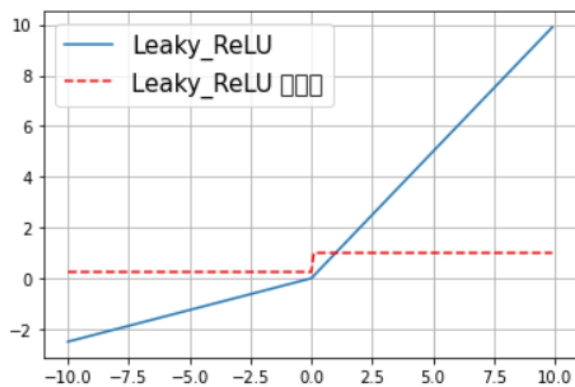


```
In [13]: def Leaky_ReLU(x, alpha):
         arr = []
         for i in x:
             arr.append(alpha*i if i<0 else i )
         return arr
```

```
In [14]: def df_Leaky_ReLU(x, alpha):
         arr = []
         for i in x:
             arr.append(alpha if i<0 else 1 )
         return arr
```

```
In [15]: x = np.arange(-10., 10., 0.1)
         alpha =0.25
         y_LeakyReLU = Leaky_ReLU(x, alpha)
         df = df_Leaky_ReLU(x, alpha)
         plt.plot(x, y_LeakyReLU, label = u'Leaky_ReLU')
         plt.plot(x, df, 'r', linestyle = '--', label = u'Leaky_ReLU 的导数')
         plt.grid()
         plt.legend(fontsize = 15)
```

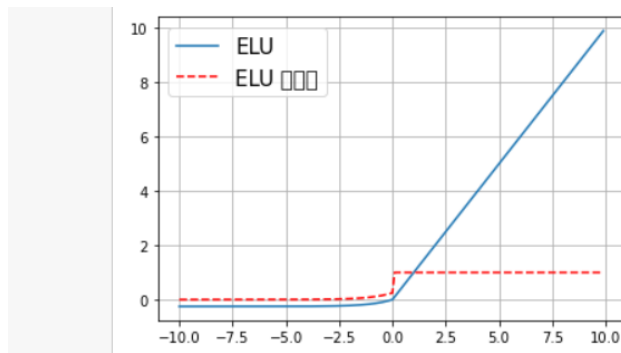
Out[15]: <matplotlib.legend.Legend at 0x1678f06a820>



```
In [16]: def ELU(x, alpha):
         arr = []
         for i in x:
             arr.append(alpha*(np.exp(i) -1) if i<0 else i )
         return arr
         def df_ELU(x, alpha):
             arr = []
             for i in x:
                 arr.append(alpha*(np.exp(i)) if i<0 else 1 )
             return arr
```

```
In [17]: x = np.arange(-10., 10., 0.1)
alpha = 0.25
y_ELU = ELU(x, alpha)
df = df_ELU(x, alpha)
plt.plot(x, y_ELU, label = u'ELU')
plt.plot(x, df, 'r', linestyle = '--', label = u'ELU 的导数')
plt.grid()
plt.legend(fontsize = 15)
```

Out[17]: <matplotlib.legend.Legend at 0x1678f08bdc0>

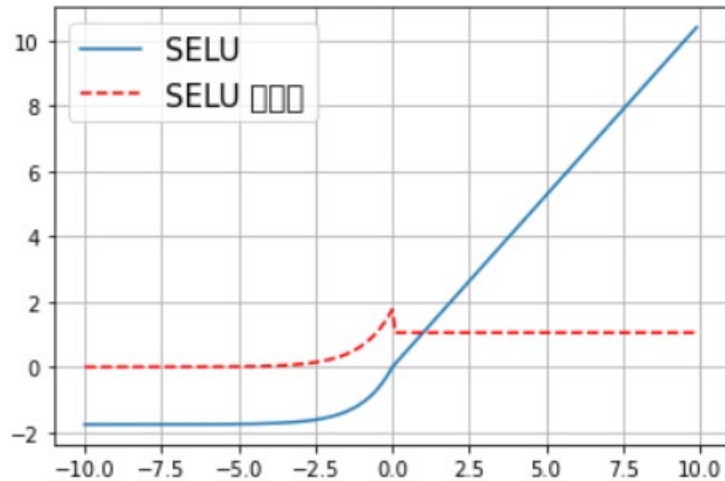


```
In [19]: def SELU(x, alpha):
arr = []
alpha = 1.6732632423543772848170429916717
lambda_ = 1.0507009873554804934193349852946
for i in x:
    arr.append(lambda_*alpha*(np.exp(i) - 1) if i < 0 else lambda_*i)
return arr
```

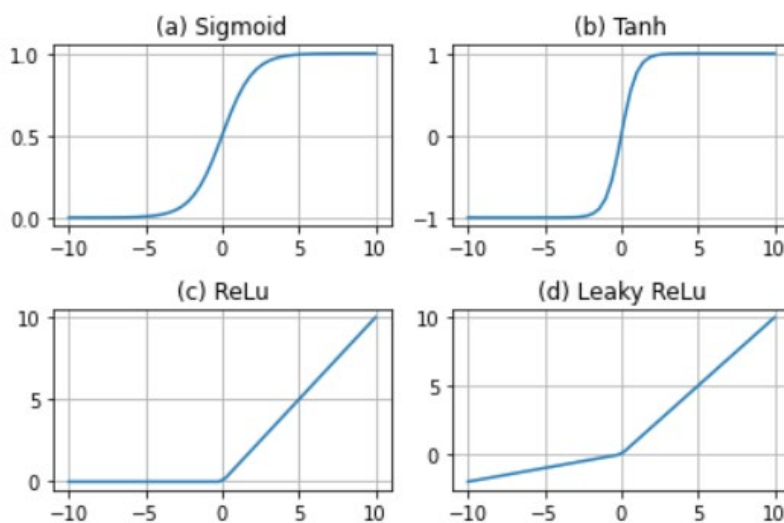
```
In [20]: def df_SELU(x, alpha):
arr = []
alpha = 1.6732632423543772848170429916717
lambda_ = 1.0507009873554804934193349852946
for i in x:
    arr.append(lambda_*alpha*(np.exp(i) ) if i < 0 else lambda_)
return arr
```

```
In [21]: x = np.arange(-10., 10., 0.1)
alpha = 0.25
y_SELU = SELU(x, alpha)
df = df_SELU(x, alpha)
plt.plot(x, y_SELU, label = u'SELU')
plt.plot(x, df, 'r', linestyle = '--', label = u'SELU 的导数')
plt.grid()
plt.legend(fontsize = 15)
```

Out[21]: <matplotlib.legend.Legend at 0x1678f05b2b0>



```
In [22]: import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(-10,10)
y_sigmoid = 1/(1+np.exp(-x))
y_tanh = (np.exp(x)-np.exp(-x))/(np.exp(x)+np.exp(-x))
fig = plt.figure()
# plot sigmoid
ax = fig.add_subplot(221)
ax.plot(x,y_sigmoid)
ax.grid()
ax.set_title(' (a) Sigmoid')
# plot tanh
ax = fig.add_subplot(222)
ax.plot(x,y_tanh)
ax.grid()
ax.set_title(' (b) Tanh')
# plot relu
ax = fig.add_subplot(223)
y_relu = np.array([0*item if item<0 else item for item in x ])
ax.plot(x,y_relu)
ax.grid()
ax.set_title(' (c) ReLu')
#plot leaky relu
ax = fig.add_subplot(224)
y_relu = np.array([0.2*item if item<0 else item for item in x ])
ax.plot(x,y_relu)
ax.grid()
ax.set_title(' (d) Leaky ReLu')
plt.tight_layout()
```

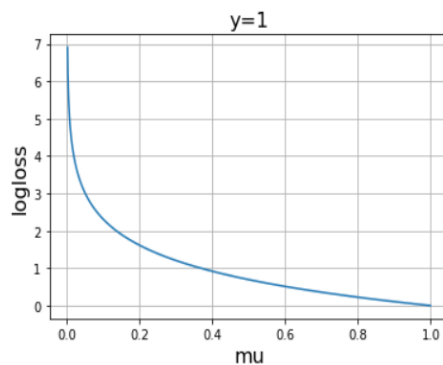


```
In [27]: import matplotlib.pyplot as plt # python 的可视化模块, 我有教程 (https://morvanzhou.github.io/tutorials/data-manipulation/p)
import torch
import torch.nn.functional as F # 激励函数都在这
from torch.autograd import Variable # 自动求导 变量
# 做一些假数据来观看图像
x = torch.linspace(-5, 5, 200) # x data (tensor), shape=(100, 1)
x = Variable(x)
x_np = x.data.numpy() # 换成 numpy array, 出图时用
# 几种常用的 激励函数
y_relu = F.relu(x).data.numpy()
y_sigmoid = F.sigmoid(x).data.numpy()
y_tanh = F.tanh(x).data.numpy()
y_softplus = F.softplus(x).data.numpy() # Softplus(x)=log(1+e^x)
plt.figure(1, figsize=(8, 6))
plt.subplot(221)
plt.plot(x_np, y_relu, c='red', label='relu')
plt.ylim((-1, 5))
plt.legend(loc='best') # 图例所有 figure 位置
plt.subplot(222)
plt.plot(x_np, y_sigmoid, c='red', label='sigmoid')
plt.ylim((-0.2, 1.2))
plt.legend(loc='best')
plt.subplot(223)
plt.plot(x_np, y_tanh, c='red', label='tanh')
plt.ylim((-1.2, 1.2))
plt.legend(loc='best')
plt.subplot(224)
plt.plot(x_np, y_softplus, c='red', label='softplus')
plt.ylim((-0.2, 6))
plt.legend(loc='best')
plt.show()
```

```
In [28]: x = np.arange(0, 1, 0.001)
logloss = -np.log( x)
plt.plot(x, logloss)
plt.grid()
plt.xlabel('mu', fontsize = 16)
plt.ylabel('logloss', fontsize = 16)
plt.title('y=1', fontsize = 16)
```

C:\Users\Administrator\AppData\Local\Temp\ipykernel\_1980\51743719.py:2: RuntimeWarning: divide by zero encountered in log  
logloss = -np.log( x)

Out[28]: Text(0.5, 1.0, 'y=1')



```
In [29]: x = np.arange(0, 1, 0.001)
logloss = -np.log(1- x)
plt.plot(x, logloss)
plt.grid()
plt.xlabel('mu', fontsize = 16)
plt.ylabel('logloss', fontsize = 16)
plt.title('y=0', fontsize = 16)
```

Out[29]: Text(0.5, 1.0, 'y=0')

