신경망(NeuralNet) ¶

In [1]:

```
pip install mglearn
```

```
Collecting mglearn
  Downloading https://files.pythonhosted.org/packages/65/38/8aced26fce0b2ae82c3c87cd
3b6105f38ca6d9d51704ecc44aa54473e6b9/mglearn-0.1.9.tar.gz (https://files.pythonhoste
d.org/packages/65/38/8aced26fce0b2ae82c3c87cd3b6105f38ca6d9d51704ecc44aa54473e6b9/mg
learn-0.1.9.tar.gz) (540kB)
     |████████████████████████████████| 542kB 8.6MB/s eta 0:00:01
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from
mglearn) (1.18.5)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.6/dist-packages
 (from mglearn) (3.2.2)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.6/dist-package
s (from mglearn) (0.22.2.post1)
Requirement already satisfied: pandas in /usr/local/lib/python3.6/dist-packages (fro
m mglearn) (1.1.2)
Requirement already satisfied: pillow in /usr/local/lib/python3.6/dist-packages (fro
m mglearn) (7.0.0)
Requirement already satisfied: cycler in /usr/local/lib/python3.6/dist-packages (fro
m mglearn) (0.10.0)
Requirement already satisfied: imageio in /usr/local/lib/python3.6/dist-packages (fr
om mglearn) (2.4.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.6/dist-packages (fro
m mglearn) (0.16.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.6/dist-pa
ckages (from matplotlib->mglearn) (1.2.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/loca
l/lib/python3.6/dist-packages (from matplotlib->mglearn) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.6/dist
-packages (from matplotlib->mglearn) (2.8.1)
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.6/dist-packag
es (from scikit-learn->mglearn) (1.4.1)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-package
s (from pandas->mglearn) (2018.9)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from c
ycler->mglearn) (1.15.0)
Building wheels for collected packages: mglearn
  Building wheel for mglearn (setup.py) ... done
  Created wheel for mglearn: filename=mglearn-0.1.9-py2.py3-none-any.whl size=582639
sha256=edd7f290bff5ac283d1b285f3bcf30f7cdf3e7f51cc7cca2a7392e08700025eb
  Stored in directory: /root/.cache/pip/wheels/eb/a6/ea/a6a3716233fa62fc561259b5cb1e
28f79e9ff3592c0adac5f0
Successfully built mglearn
Installing collected packages: mglearn
Successfully installed mglearn-0.1.9
```

In [3]:

```
import mglearn
```

In [4]:

```
mglearn.plots.plot_logistic_regression_graph()
```

Out[4]:

```
<graphviz.dot.Digraph at 0x7f7f8e053dd8>
```

- 왼쪽 노드는 입력 특성
- 연결선은 학습된 계수
- 오른쪽 노드는 입력의 가중치의 합, 즉 출력

은닉층이 하나인 다층 퍼셉트론

In [5]:

```
mglearn.plots.plot_single_hidden_layer_graph()
```

Out[5]:

```
<graphviz.dot.Digraph at 0x7f7f739c1550>
```

활성화 함수

Relu(렐루-rectified linear unit, ReLU), tanh(하이퍼 볼릭 탄젠트-hyperbolic tangent)

- ReLU 함수는 0이하를 잘라버림.
- tanh 함수는 낮은 입력값에 대해 -1로 수렴, 큰 입력값에 대해 +1로 수렴
- sigmoid 함수는 낮은 입력값에 대해 0에 수렴, 큰 입력값에 대해 1로 수렴

In [6]:

```
import numpy as np
import matplotlib.pyplot as plt
```

In [7]:

```
line = np.linspace(-3, 3, 100)
tanh_line = np.tanh(line)
relu_line = np.maximum(line, 0)  # 두개의 배열값 중 최대값 찾기
sig_line = 1/(1+np.exp(-line))

step_line = line.copy()
step_line[step_line <= 0] = 0
step_line[step_line > 0] = 1
```

In [9]:

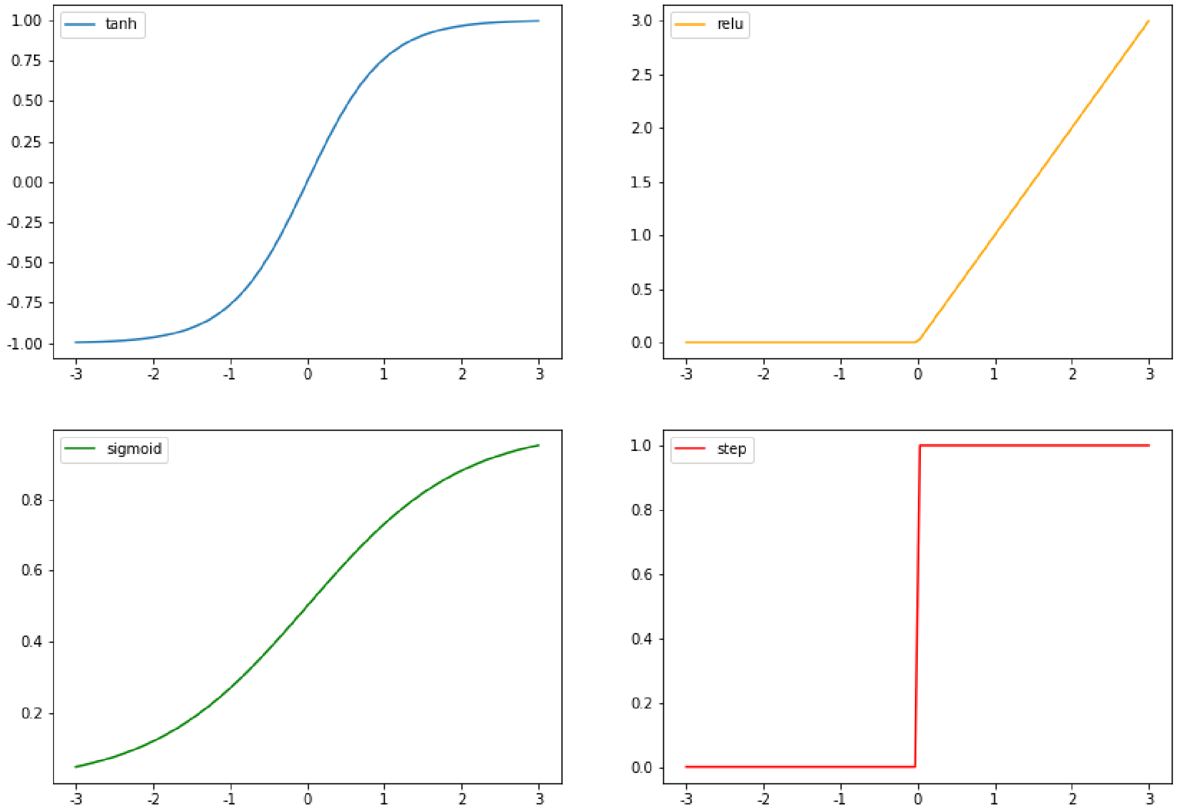
```
# 음수 표시
import matplotlib
matplotlib.rcParams['axes.unicode_minus'] = False
plt.rcParams["figure.figsize"] = (14,10)
```

In [17]:

```
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2)
ax1.plot(line, tanh_line, label='tanh')
ax1.legend(loc='best')
ax2.plot(line, np.maximum(line, 0), label='relu', color='orange')
ax2.legend(loc='best')
ax3.plot(line, sig_line, label='sigmoid', color='green')
ax3.legend(loc='best')
ax4.plot(line, step_line, label='step', color='red')
ax4.legend(loc='best')
```

Out[17]:

```
<matplotlib.legend.Legend at 0x7f7f611deda0>
```

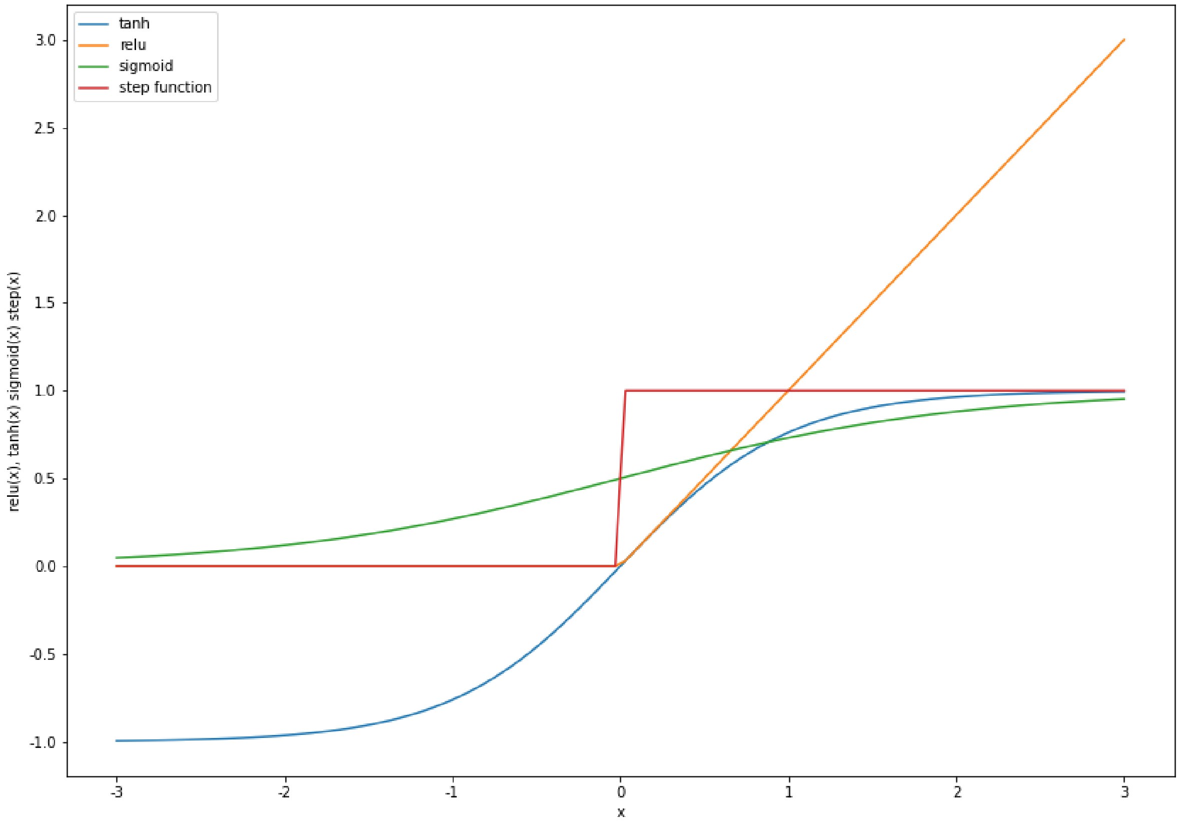


In [18]:

```
plt.plot(line, tanh_line, label='tanh')
plt.plot(line, relu_line, label='relu')
plt.plot(line, sig_line, label='sigmoid')
plt.plot(line, step_line, label='step function')
plt.legend(loc='best')
plt.xlabel('x')
plt.ylabel('relu(x), tanh(x) sigmoid(x) step(x)')
```

Out[18]:

```
Text(0, 0.5, 'relu(x), tanh(x) sigmoid(x) step(x)')
```



In [19]:

```
mglearn.plots.plot_two_hidden_layer_graph()
```

Out[19]:

```
<graphviz.dot.Digraph at 0x7f7f613673c8>
```

In [ ]:

```

```