10.1.38.19/user/ra2311047010027/lab/tree/Deep%20Learning/dl5.ipynb

ernel Tabs Settings Help

dl4.ipynb dl5.ipynb

Code Notebook

Modified
days ago
days ago
inutes ago
inutes ago

[1]:
```
import numpy as np
import matplotlib.pyplot as plt

# Activation functions
def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def tanh(x):
    return np.tanh(x)

def relu(x):
    return np.maximum(0, x)

def softmax(x):
    exp_x = np.exp(x - np.max(x))
    return exp_x / exp_x.sum()

# Input values
x = np.linspace(-10, 10, 100)

# Apply activation functions
y_sigmoid = sigmoid(x)
y_tanh = tanh(x)
y_relu = relu(x)
y_softmax = softmax(x)
```

10.1.38.19/user/ra2311047010027/lab/tree/Deep%20Learning/dl5.ipynb

Incognito

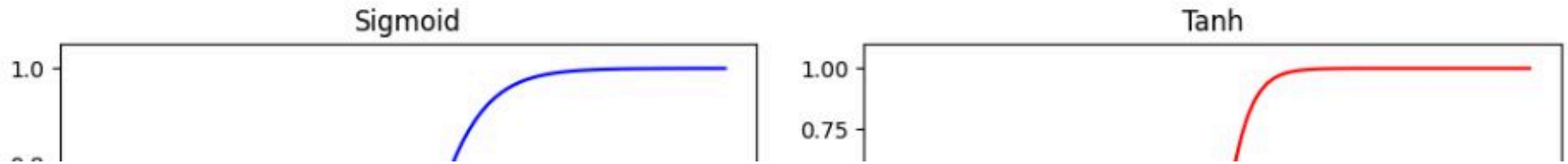
rnel Tabs Settings Help

dl4.ipynb dl5.ipynb

Code Notebook Python 3 (ipykernel)

Modified
days ago
days ago
utes ago
utes ago

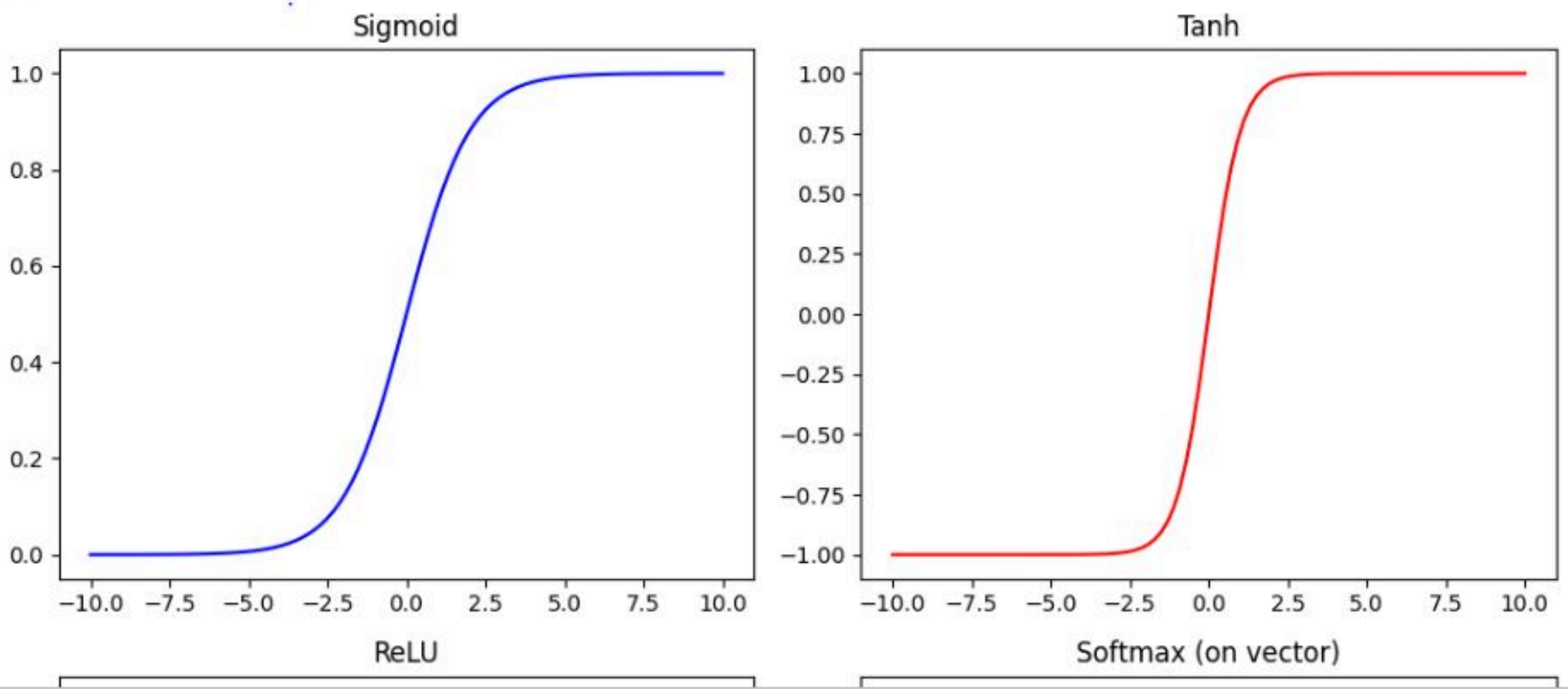
```
# Plot
plt.figure(figsize=(10,8))

plt.subplot(2,2,1)
plt.plot(x, y_sigmoid, 'b')
plt.title("Sigmoid")

plt.subplot(2,2,2)
plt.plot(x, y_tanh, 'r')
plt.title("Tanh")

plt.subplot(2,2,3)
plt.plot(x, y_relu, 'g')
plt.title("ReLU")

plt.subplot(2,2,4)
plt.plot(x, y_softmax, 'm')
plt.title("Softmax (on vector)")

plt.tight_layout()
plt.show()
```



ipynb | dl5.ipynb

Code | Notebook | Python 3 (ipykernel)

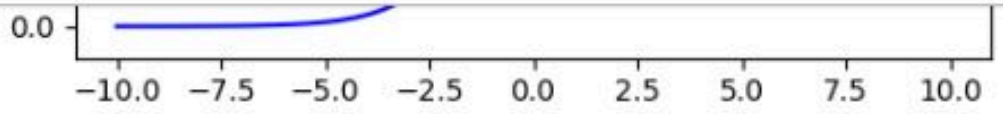
```
plt.title("Softmax (on vector)")

plt.tight_layout()
plt.show()
```

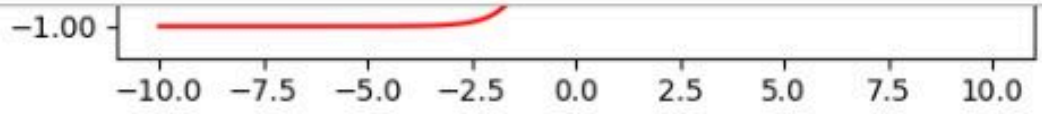


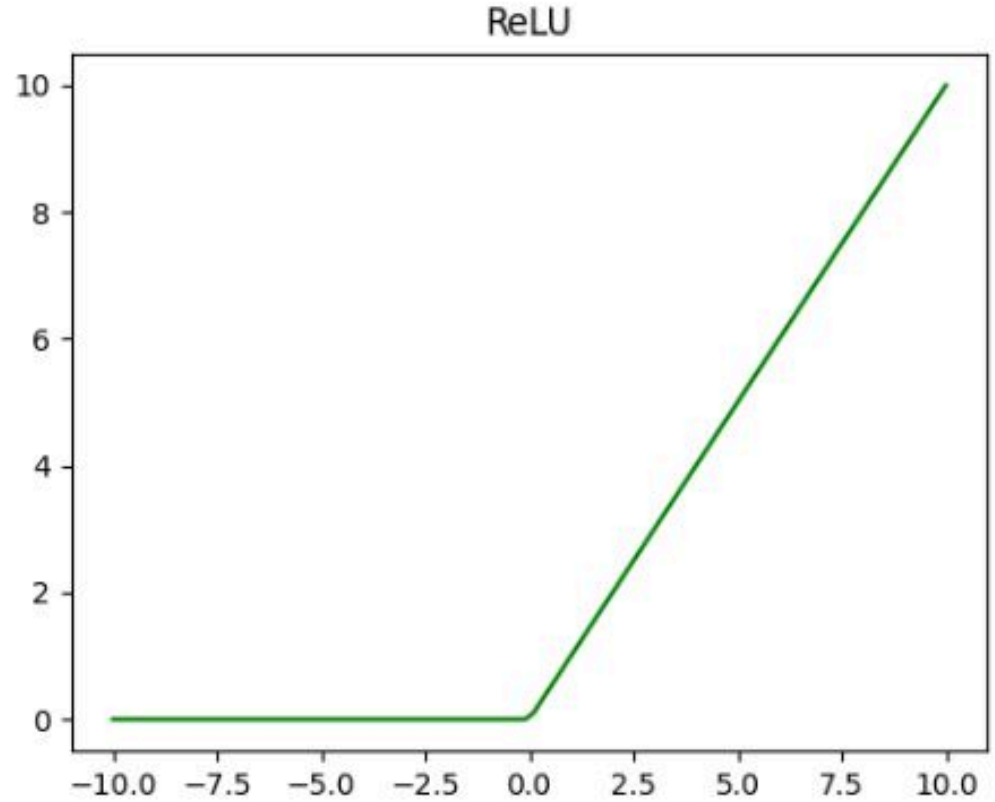
tings Help

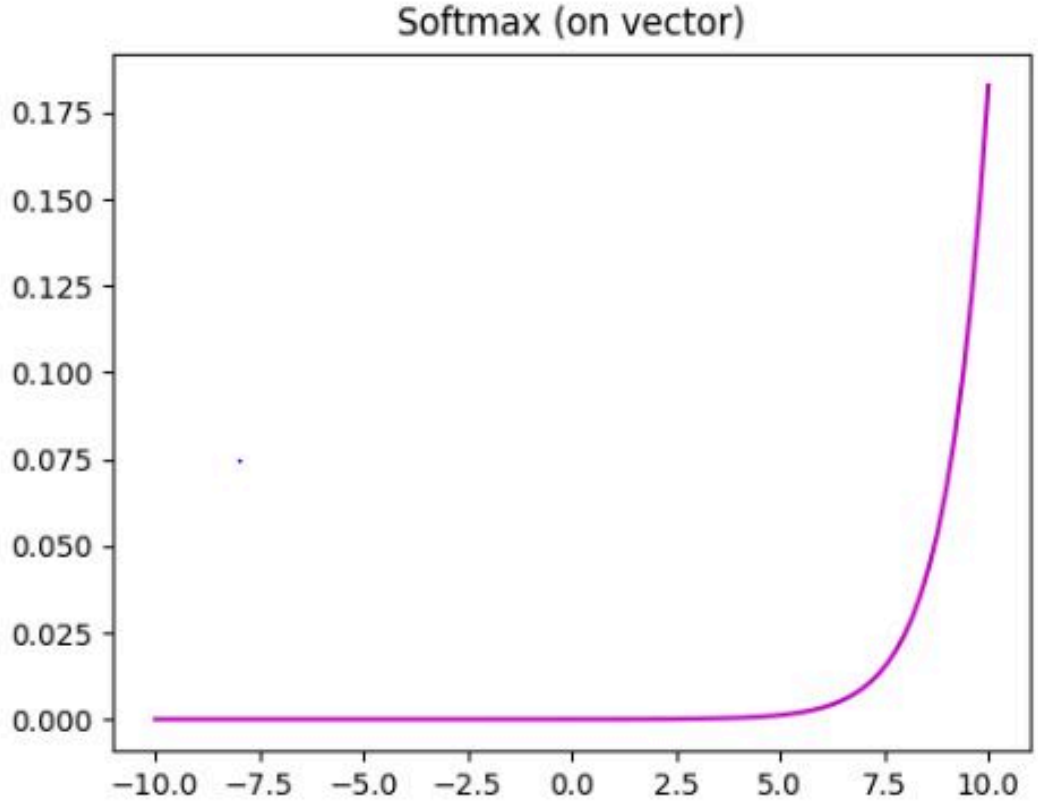
dl4.ipynb dl5.ipynb

Code Notebook Python 3 (ipykernel)









[ ]:

01/08/25 # 5. Study of Activation functions and its role

Aim :

To study different activation functions and understand their role.

Objective :

1. To implement and visualize common activation functions such as Sigmoid, Tanh, ReLU and softmax.
2. To understand the importance of activation function in transforming input signals.
3. To observe how activation functions affect the training & performance of models.

Pseudocode :

non-linear ?

Start

1. Import necessary libraries (numpy, matplotlib, etc.)
2. Define activation functions
   (i) Sigmoid(x) = 1/(1 + e^(-x)) = [0, 1]
   (ii) Tanh(x) = (e^x - e^(-x))/(e^x + e^(-x)) [-1, 1]
   (iii) ReLU(x) = max(0, x) [0, ∞]
   (iv) softmax(xi) = e^(xi) / Σ e^(xj) for all j
3. Generate input values in a range (eg: -10 to 10)
4. Apply each activation function on the input values.
5. plot the outputs of each function to visualize their behaviour.

6. Compare and observe :
   - Range of outputs
   - Non-linearity introduced
   - Suitability for classification / regression

End

Observation :

1. Sigmoid : Squashes values between 0 and 1. useful for probabilities but suffers from vanishing gradients.

2. Tanh : Squashes values between -1 and 1, centered at zero, better than sigmoid in some cases

3. ReLU : Output 0 for negative inputs and linear for positive, avoids vanishing gradients Commonly used in hidden layers

4. Softmax : Converts outputs into probabilities that sum to 1, mainly used in final layer of classification tasks.

Result :

Successfully studied and implemented activation functions and understood their role.