From Paper to Code: A Guide to Implement Research

Ahad Jawaid

2023-06-05

::: {.cell 0=‘h’ 1=‘i’ 2=‘d’ 3=‘e’}

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

:::

::: {.cell 0=‘h’ 1=‘i’ 2=‘d’ 3=‘e’}

```
def graph_fn(fn, fig=None, ax=None, xlim=(-10,10), ylim=(-10,10), size=(6, 6)):
    if fig is None or ax is None:
        fig, ax = plt.subplots(figsize=size)

    x = np.linspace(xlim[0], xlim[1], 400)
    y = fn(x)
    ax.plot(x, y)

    ax.axhline(0, color='black',linewidth=0.5)
    ax.axvline(0, color='black',linewidth=0.5)

    ax.set_xlabel('x')
    ax.set_ylabel('f(x)')
    ax.set_title(fn.__name__)

    if xlim:
        ax.set_xlim(xlim)
    if ylim:
        ax.set_ylim(ylim)
    ax.grid(True)

    return fig, ax
```

:::

Activation Functions

An activation function transforms it’s inputs to an output that is nonlinear. A nonlinear function is a function that transform the inputs such that the outputs are not scaled by a single number for example a stright line on a graph. An example of a nonlinear function is a line that is clamped at 0 as show in the following figure:

::: {.cell 0=‘h’ 1=‘i’ 2=‘d’ 3=‘e’}

```
def Linear(x):
    return x
```

:::

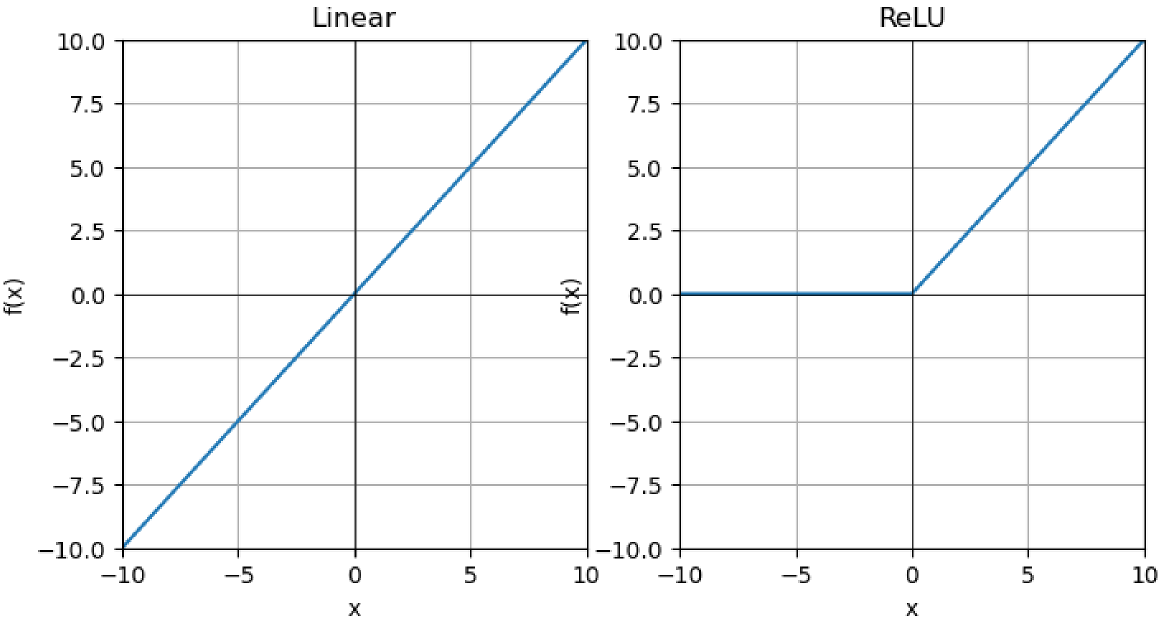
::: {.cell 0=‘h’ 1=‘i’ 2=‘d’ 3=‘e’}

```
def ReLU(x):
    return np.maximum(0, x)
```

:::

::: {.cell 0=‘h’ 1=‘i’ 2=‘d’ 3=‘e’}

```
fns = [Linear, ReLU]
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(8,4))
for ax, fn in zip(axs, fns): graph_fn(fn, ax=ax, fig=fig, ylim=(-10, 10));
```


Linear
ReLU
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
f(x)
−10
−5
0
5
10
x

:::