

Forward propagation

Puteaux, Fall/Winter 2020-2021

```
#####  
##                               ##  
##  Deep Learning in Python  ##  
##                               ##  
#####
```

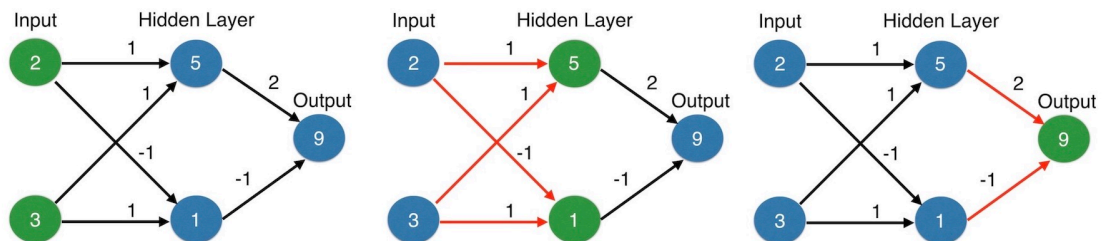
§1 Introduction to Deep Learning in Python

§1.1 Basics of deep learning and neural networks

§1.1.2 Forward propagation

1. How does the forward propagation function?

- Multiply - add process.
- Dot product.
- Forward propagation is for one data point at a time.
- The output is the prediction for that data point.



2. Code of the forward propagation:

```
[1]: import numpy as np  
  
input_data = np.array([2, 3])  
weights = {  
    'node_0': np.array([1, 1]),  
    'node_1': np.array([-1, 1]),  
    'output': np.array([2, -1])  
}
```

```

node_0_value = (input_data * weights['node_0']).sum()
node_1_value = (input_data * weights['node_1']).sum()

hidden_layer_values = np.array([node_0_value, node_1_value])
print(hidden_layer_values)

```

[5 1]

```

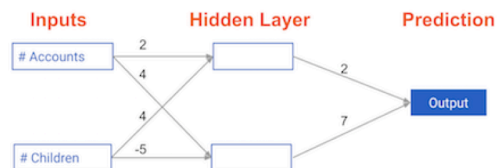
[2]: output = (hidden_layer_values * weights['output']).sum()
print(output)

```

9

3. Practice exercises for the forward propagation:

► Diagram of the forward propagation:



► Package pre-loading:

```

[3]: import numpy as np

```

► Data pre-loading:

```

[4]: input_data = np.array([3, 5])

weights = {
    'node_0': np.array([2, 4]),
    'node_1': np.array([4, -5]),
    'output': np.array([2, 7])
}

```

► The forward propagation algorithm practice:

```

[5]: # Calculate node 0 value: node_0_value
node_0_value = (input_data * weights['node_0']).sum()

# Calculate node 1 value: node_1_value
node_1_value = (input_data * weights['node_1']).sum()

# Put node values into array: hidden_layer_outputs
hidden_layer_outputs = np.array([node_0_value, node_1_value])

# Calculate output: output

```

```
output = (hidden_layer_outputs * weights['output']).sum()

# Print output
print(output)
```

-39