**Name:** Rafid Redwan Khan

**ID:** 20-42239-1

**Section:** A

**Report on Activation Function**

An activation function in a neural network is a mathematical function that establishes a neuron's output in accordance with the input it receives. The activation function, which provides non-linearity into the model and enables it to learn more complicated patterns and relationships in the data, is a crucial part of the artificial neural network. There are six different activation functions that are described below,

1. **Step Function**

Binary step function returns value either 0 or 1.

* It returns '0' if the input is the less than zero
* It returns '1' if the input is greater than zero

**If x >= 0, then f(x) = 1.**

**f(x) = 0, if x < 0**

**Advantage:**

* Step Functions let us easily link services, systems, or individuals by describing intricate business logic with low-code, event-driven processes.
* It is simple to use
* It gives us discrete output

**Disadvantage:**

* It has a tendency of vanishing gradient
* It has a larger learning curve

1. **Sigmoid Function**

Sigmoid function returns the value between 0 and 1. For activation function in deep learning network, Sigmoid function is considered not good since near the boundaries the network doesn't learn quickly.

**1 / (1 + e(-x)) f(x)**

**Advantage:**

* It provides smooth gradient which helps us in preventing “jumps” in output values.
* It is suitable for binary classification problems.

**Disadvantage:**

* The outputs are not zero centered
* It is computationally expensive

1. **Tanh Function**

Tanh is another nonlinear activation function. Tanh outputs between -1 and 1. Tanh also suffers from gradient problem near the boundaries just as Sigmoid activation function does.

**f(x) is calculated as (ex - e(-x)) / (ex + e(-x)).**

**Advantage:**

* As the function is differentiable, the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph.

**Disadvantage:**

* The output of the tanh function is not centered around zero, which can make optimization more challenging.

1. **Relu Function**

RELU is more well-known activation function which is used in the deep learning networks. RELU is less computational expensive than the other nonlinear activation functions.

**max(f(x) (0, x)**

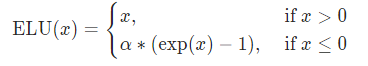
**Advantage:**

* It does not activate all the neurons at the same time.
* This function is non-linear and computationally efficient

**Disadvantage:**

* It cannot learn on examples for which their activation is zero.
* It gives unbounded output
* It is not suitable for negative inputs

1. **Elu Function**

The Exponential Linear Unit (ELU) is an activation function commonly used in artificial neural networks. The ELU activation function is a differentiable function that maps the input values to an output range of (-1, ∞).

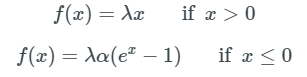
**Advantage:**

* The ELU function can produce negative values, unlike ReLU
* The ELU function has a mean output closer to zero compared to ReLU

**Disadvantage:**

* The ELU function involves computing exponentials, which can make it slower to compute.
* The exponential function used in the ELU function can potentially lead to numerical instability if the input is very large or very small, which can cause problems during training.

1. **Selu**

The SELU activation function is a variant of the Exponential Linear Unit (ELU) activation function that has been shown to work well in deep neural networks.

**Advantage:**

* Like ReLU, SELU does not have vanishing gradient problem
* SELUs learn faster and better than other activation functions without needing further procession.

**Disadvantage:**

* SELU is a relatively new activation function so it is not yet used widely in practice. ReLU stays as the preferred option.
* The computation of the SELU function involves exponential and logarithmic operations, which can make it computationally more expensive.