

---

Certified Professional in AI Applications in Aviation

## Neural Networks and Deep Learning for Aviation

---

### **\*\*Activation Function\*\***

A function applied to the output of a neural network node (neuron) that determines whether it will be activated or not, based on its weighted input. Common activation functions include the step function, sigmoid function, tanh function, and rectified linear unit (ReLU) function.

### **\*\*Artificial Intelligence (AI)\*\***

The simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using rules to reach approximate or definite conclusions), and self-correction.

### **\*\*Backpropagation\*\***

A supervised learning algorithm used to train neural networks by minimizing the error between the predicted and actual outputs. It involves propagating the error backwards through the network, adjusting the weights of the connections between the nodes to reduce the error.

### **\*\*Convolutional Neural Network (CNN)\*\***

A type of neural network commonly used for image processing and computer vision tasks. CNNs use convolutional layers, which apply a set of filters to the input data to extract features such as edges, shapes, and textures.

### **\*\*Deep Learning\*\***

A subset of machine learning that uses artificial neural networks with many layers (deep structures) to learn and represent data. Deep learning models can automatically learn complex features and patterns from large datasets, and are widely used in applications such as image and speech recognition, natural language processing, and autonomous systems.

### **\*\*Epoch\*\***

A single pass through the entire training dataset during the training of a neural network. An epoch is completed when all of the samples have been used once to update the network's weights.

### **\*\*Fully Connected Layer\*\***

A layer in a neural network where every node in the layer is connected to every node in the previous and next layers. Also known as a dense layer.

**\*\*Gradient Descent\*\***

An optimization algorithm used to minimize the error or loss function in machine learning and neural networks. Gradient descent involves iteratively adjusting the weights and biases in the direction of the negative gradient of the error function with respect to the weights and biases.

**\*\*Hyperparameter\*\***

A parameter that is set before the training of a machine learning model, and is not learned from the data. Examples of hyperparameters include the learning rate, the number of hidden layers, and the number of nodes in each layer.

**\*\*Input Layer\*\***

The first layer in a neural network, which receives the raw input data.

**\*\*Learning Rate\*\***

A hyperparameter that determines the step size at each iteration while moving toward a minimum of a loss function. A learning rate that is too small may result in a long training time, while a learning rate that is too large may cause the model to converge to a suboptimal solution.

**\*\*Long Short-Term Memory (LSTM)\*\***

A type of recurrent neural network (RNN) that is capable of learning long-term dependencies in sequential data. LSTMs use special units called memory cells to store and access information over long periods of time, and have been used in applications such as speech recognition, machine translation, and language modeling.

**\*\*Mean Squared Error (MSE)\*\***

A common loss function used in regression problems, where the goal is to predict a continuous output variable. MSE is the average of the squared differences between the predicted and actual outputs.

**\*\*Neural Network\*\***

A computational model inspired by the structure and function of the human brain. A neural network consists of interconnected nodes (neurons) that process and transmit information. Neural networks can be trained to perform a variety of tasks, such as classification, regression, and prediction.

**\*\*Output Layer\*\***

The final layer in a neural network, which produces the output of the model.

**\*\*Overfitting\*\***

A situation in machine learning where a model learns the training data too well, capturing the noise and

random fluctuations in the data, and performs poorly on new, unseen data. Overfitting can be reduced by using regularization techniques, such as dropout and weight decay.

#### **\*\*Recurrent Neural Network (RNN)\*\***

A type of neural network that is designed to handle sequential data, such as time series or natural language. RNNs have a feedback connection that allows information from previous time steps to be used in the current time step.

#### **\*\*Regularization\*\***

A technique used to prevent overfitting in machine learning models by adding a penalty term to the loss function. Regularization encourages the model to have smaller weights and biases, and reduces the complexity of the model.

#### **\*\*Sigmoid Function\*\***

A smooth, S-shaped activation function that maps any input value to a value between 0 and 1. The sigmoid function is often used in the output layer of a binary classification problem.

#### **\*\*Supervised Learning\*\***

A type of machine learning where the model is trained on labeled data, i.e., data with known input-output pairs. The goal of supervised learning is to learn a mapping from inputs to outputs that can be used to make predictions on new, unseen data.

#### **\*\*Tanh Function\*\***

A smooth, S-shaped activation function that maps any input value to a value between -1 and 1. The tanh function is similar to the sigmoid function, but is centered around 0, which can make it easier to train deep neural networks.

#### **\*\*Unsupervised Learning\*\***

A type of machine learning where the model is trained on unlabeled data, i.e., data without known input-output pairs. The goal of unsupervised learning is to discover patterns and structure in the data, such as clusters, dimensions, or distributions.

#### **\*\*Weight\*\***

A parameter in a neural network that determines the strength of the connection between two nodes. Weights are learned during the training process, and represent the importance of the input features in the model.

#### **\*\*Weight Decay\*\***

A regularization technique that adds a penalty term to the loss function, proportional to the size of the

---

weights. Weight decay encourages the model to have smaller weights, and reduces the risk of overfitting.

**\*\*Xavier Initialization\*\***

A technique for initializing the weights in a neural network, which aims to ensure that the variance of the output of each node is similar to the variance of its input. Xavier initialization is also known as Glorot initialization.