Slide 1

The diagram shows a simple sketch of an Artificial Neural Network (ANN). An ANN is a type of machine learning model that is inspired by the structure and function of the human brain. ANNs are composed of layers of interconnected nodes, or neurons, that can process and transmit information.

In the diagram, the input layer is on the left-hand side. This layer consists of three independent variables, x1, x2, and x3. These variables represent the input data that is fed into the ANN.

The middle layer is the summing junction. This is where the input variables are weighted and summed together. The weights represent the strength of the connection between each input variable and the neuron in the summing junction.

The output layer is on the right-hand side. This layer consists of one dependent variable, y. The dependent variable represents the output of the ANN. The output is determined by the sum of the weighted input variables and the activation function of the neuron in the summing junction.

The activation function is a mathematical function that determines the output of a neuron. The activation function is typically a non-linear function, which means that the output is not simply a linear combination of the inputs. This allows ANNs to learn complex patterns in data.

Slide 2

**Parallel processing**  
One of the major advantages of the neural network is its ability to do many things at once. With traditional computers, processing is sequential--one task, then the next, then the next, and so on. The idea of threading makes it appear to the human user that many things are happening at one time.

The artificial neural network is an inherently multiprocessor-friendly architecture. Without much modification, it goes beyond one or even two processors of the von Neumann architecture. The artificial neural network is designed from the onset to be parallel. Humans can listen to music at the same time they do their homework--at least, that's what we try to convince our parents in high school. With a massively parallel architecture, the neural network can accomplish a lot in less time. The tradeoff is that processors have to be specifically designed for the neural network.

**The ways in which they function**  
Think of traditional computers like really smart rule-followers. They understand instructions and do calculations, but they need specific rules to work. It's like teaching a robot to follow a recipe step by step.

Now, imagine artificial neural networks as more creative learners. They don't just follow rules; they learn from examples and experiences. It's like teaching a kid to ride a bike by letting them try and figure it out, rather than giving them a manual.

So, traditional computers need clear rules, while neural networks learn from doing things. This makes them good for different jobs.

**Self-programming**  
Traditional computers, like the von Neumann ones, need someone to tell them what to do using languages like C or Java. It's like giving them a recipe to follow. They can't change the recipe on their own.

Now, picture artificial neural networks as the creative chefs of the computing world. They don't just follow recipes; they can change and improve their own cooking methods. They kinda teach themselves new recipes!

**Speed**  
When it comes to speed, traditional computers need big processors or the complex idea of parallel processors (imagine doing many things at once). On the other hand, neural networks need special custom-built chips for their tasks.

So, traditional computers follow instructions, while neural networks can kinda invent their own and need special tools for their cooking. Cool, right?

**Slide 3**

**Neurons**

The basic unit of a neural network is a neuron. A neuron receives input from other neurons or from an external source and produces an output based on that input. The output of a neuron is determined by its activation function, which is a mathematical function that maps the input to an output. The activation function can be a simple threshold function, a sigmoid function, a ReLU function, or any other function that is appropriate for the problem at hand.

**Layers**

A neural network consists of multiple layers of neurons. The input layer is the first layer of the network, which receives input data from an external source. The output layer is the last layer of the network, which produces the final output of the network. The hidden layers are the layers in between the input and output layers, which perform computations on the input data.

**Weights and Bias**

Each connection between neurons in adjacent layers is associated with a weight. The weight determines the strength of the connection between the neurons. The weights are updated during the training process of the neural network to minimize the error between the predicted output and the actual output. In addition to weights, each neuron in a neural network has a bias term, which is a constant value that is added to the input of the neuron. The bias term allows the neuron to adjust its output independently of the input.

**Training**

The process of training a neural network involves adjusting the weights and biases of the neurons to minimize the error between the predicted output and the actual output. This is typically done using an optimization algorithm such as stochastic gradient descent. During training, the neural network learns to recognize patterns in the input data and make predictions based on those patterns.

**Deep Learning**

Deep Learning is a subfield of Machine Learning that is based on the use of deep neural networks. Deep neural networks are neural networks that have multiple hidden layers. The use of multiple hidden layers allows the network to learn more complex representations of the input data, which can lead to better performance on a wide range of tasks.

**Slide 4 - End**

So, in simple terms, you're giving the computer ingredients (inputs), deciding their importance (weights), mixing them up (transfer function), adding complexity (activation function), adjusting taste (bias), and creating a team of chefs (neural network layers). Voila, you've got the recipe for a smart cooking computer!

#### **Input Layer**

The data that we feed to the model is loaded into the input layer from external sources like a CSV file or a web service. It is the only visible layer in the complete Neural Network architecture that passes the complete information from the outside world without any computation.

#### **Hidden Layers**

The hidden layers are what makes deep learning what it is today. They are intermediate layers that do all the computations and extract the features from the data.

There can be multiple interconnected hidden layers that account for searching different hidden features in the data. For example, in image processing, the first hidden layers are responsible for higher-level features like edges, shapes, or boundaries. On the other hand, the later hidden layers perform more complicated tasks like identifying complete objects (a car, a building, a person).

#### **Output Layer**

The output layer takes input from preceding hidden layers and comes to a final prediction based on the model’s learnings. It is the most important layer where we get the final result.

In the case of classification/regression models, the output layer generally has a single node. However, it is completely problem-specific and dependent on the way the model was built.