I found the following definitions and explanations for terms and concepts in this document.

**CPUs, T4 GPUs, and TPU v2-8s**

* These are different types of processors with varying strengths and weaknesses. CPUs are general-purpose processors, T4 GPUs are parallel processors well-suited for deep learning, and TPU v2-8s are custom-designed processors specifically for machine learning and AI workloads.

**Input Layer**

* The input layer is the first layer of the neural network. It receives the input data and passes it to the next layer.

**Hidden Layers**

* Hidden layers are intermediate layers between the input and output layers. They perform computations and learn representations of the data.

**Output Layer**

* The output layer produces the final result or prediction of the neural network.

**Neurons**

* Neurons are the fundamental processing units of a neural network. They receive input, perform a computation, and produce an output.

**Training Loop**

* The training loop is the iterative process of feeding data to the neural network, calculating the error, and updating the network's parameters to improve its performance.

**Forward Pass**

* The forward pass is the process of passing the input data through the network to generate a prediction.

**Loss Calculation**

* The loss calculation involves determining the difference between the predicted output and the actual target output.

**Backward Pass and Optimization**

* The backward pass calculates the gradients of the loss with respect to the network's parameters. The optimizer then updates these parameters to minimize the loss.

**Epochs**

* An epoch refers to one complete pass through the entire training dataset during the training process.

**Predictions**

* Predictions are the outputs generated by the neural network when given new input data.

torch.Tensor

* In PyTorch, a torch.Tensor is a multi-dimensional array that serves as the fundamental building block for all operations and models. It is similar to a NumPy array but with added functionalities for deep learning, such as GPU support and automatic differentiation.

**Automatic Differentiation**

* Automatic differentiation is a technique used to automatically calculate the gradients of functions, which is crucial for training neural networks.

torch.nn **Module**

* The torch.nn module in PyTorch provides a collection of pre-built layers, activation functions, and other components for constructing neural networks.

**Loss Function**

* A loss function measures the difference between the network's predictions and the actual target values, quantifying the error the network makes.

nn.MSELoss()

* This creates an instance of the MSELoss class from PyTorch's nn module, representing the Mean Squared Error Loss, commonly used for regression tasks.

**Optimizer**

* An optimizer is an algorithm that adjusts the network's parameters (weights and biases) to minimize the loss function.

optim.SGD(...)

* This creates an instance of the SGD class from PyTorch's optim module, representing Stochastic Gradient Descent, a widely used optimization algorithm.

**Learning Rate (**lr**)**

* The learning rate controls the step size taken in the direction of the negative gradient during each iteration of the optimization process.