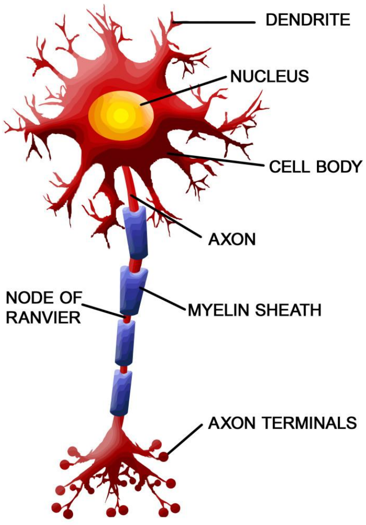
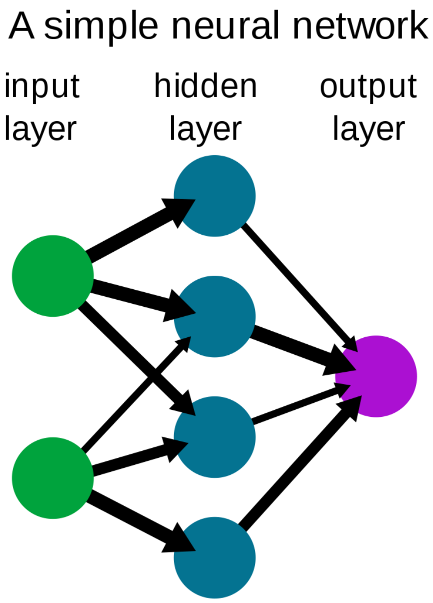
Answer_1:

A neuron is the most basic computational unit of any neural network, including the brain. It is composed of three parts: the dendrites, the cell body (soma), and the axon. The dendrites receive signals from other neurons or from sensory receptors. The cell body processes these signals and generates an output signal that is transmitted along the axon to other neurons or to an effector organ such as a muscle or gland.



A neural network is a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection (synapse) between neurons can transmit a signal from one neuron to another. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it.

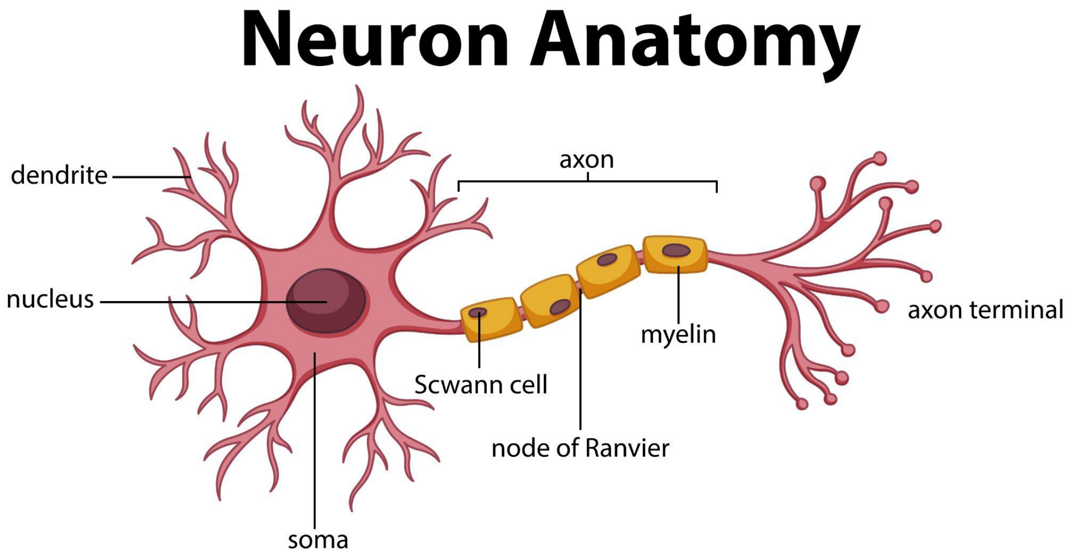


So, in summary, a neuron is a single unit that receives input and produces output while a neural network is a collection of interconnected neurons that work together to process input and produce output

Answer_2:

A neuron is composed of three parts: the dendrites, the cell body (soma), and the axon. The dendrites receive signals from other neurons or from sensory receptors. The cell body processes these signals and generates an output signal that is transmitted along the axon to other neurons or to an effector organ such as a muscle or gland.

The total number of neurons in the brain of a human is around a hundred billion. On average, each neuron may be connected to up to 10,000 other neurons, passing signals to each other via as many as 1,000 trillion synapses.



Answer_3:

A perceptron is a machine learning algorithm for the supervised learning of binary classifiers. It consists of four parts: an input value or one input layer, weights and bias, net sum, and an activation function. The input layer of the perceptron is made of artificial input neurons and takes the initial data into the system for further processing. The weight coefficient is automatically learned. Initially, weights and input features are multiplied. It calculates the total sum. A neuron can be activated or not, and is determined by an activation function. The activation function in any perceptron learning algorithm example employs a step rule to determine whether the weight function’s value is higher than zero.

Answer_4:

A single-layer perceptron is a neural network that consists of one input layer and one output layer. A multi-layer perceptron is a neural network that contains one or more hidden layers in addition to the input and output layers. A single-layer perceptron can only learn linear functions, while a multi-layer perceptron can also learn non-linear functions. Therefore, a multi-layer perceptron is more suitable for complex and non-linear problems.

Answer_5:

Forward propagation is a technique used to find the actual output of neural networks. It involves feeding the input to the network in a forward direction and calculating the pre-activation and activation values at each layer. Forward propagation is performed in a feed-forward network, which is a network configuration that does not have any loops or cycles. Forward propagation helps to decide whether the assigned weights are good to learn for the given problem statement.

Answer_6:

Backpropagation is a widely used method for calculating derivatives inside deep feedforward neural networks. It is an algorithm for supervised learning of artificial neural networks using gradient descent. Given an artificial neural network and an error function, the method calculates the gradient of the error function with respect to the neural network's weights. This gradient is then used to update the weights in such a way as to minimize the error function.

Answer_7:

The chain rule is used in backpropagation to calculate the gradient of the loss function for a single weight. It allows us to find the derivative of composite functions. The backpropagation algorithm computes the gradient of the loss function for a single weight by the chain rule. It efficiently computes one layer at a time, unlike a native direct computation. It computes the gradient, but it does not define how the gradient is used.

Answer_8:

A loss function is a mathematical function that calculates the difference between the predicted output of a neural network and the actual output. Loss functions are used to measure how well the network is performing on a particular task, such as image classification or language translation. *Common loss functions used in neural networks include:*

1. *Mean Absolute Error (MAE)*
2. *Mean Squared Error (MSE)*
3. *Huber Loss*
4. *Cross-Entropy Loss (a.k.a Log loss)*
5. *Relative Entropy (Kullback–Leibler divergence)*
6. *Squared Hinge.*

The loss function in a neural network quantifies the difference between the expected outcome and the outcome produced by the machine learning model. From the loss function, we can derive the gradients which are used to update the weights. The average over all losses constitutes the cost.

Answer_9:

Here are some examples of loss functions used in neural networks:

1. ***Mean Absolute Error (MAE):*** used for regression problems.
2. ***Mean Squared Error (MSE):*** also used for regression.
3. ***Huber Loss:*** typically used in regression problems.
4. ***Cross-Entropy Loss (a.k.a Log loss):*** used for classification problems.
5. ***Relative Entropy (Kullback–Leibler divergence)***
6. ***Squared Hinge***
7. *Maximum Likelihood provides a framework for choosing a loss function when training neural networks and machine learning models in general.*

Answer_10:

Optimizers are algorithms or methods used to change the attributes of the neural network such as weights and learning rate to reduce the losses. They are used to solve optimization problems by minimizing the function. Different optimizers are used in neural networks in order to minimize the error in predictions made by the network. The most popular optimizers are stochastic gradient descent (SGD), Adam, and RMSprop.

An optimizer is a function or an algorithm that adjusts the attributes of the neural network, such as weights and learning rates. Thus, it helps in reducing the overall loss and improving accuracy. The problem of choosing the right weights for the model is a daunting task, as a deep learning model generally consists of millions of parameters.

Answer_11:

The exploding gradient problem is a problem in the training of neural networks where the gradients used to update the weights grow exponentially. This can result in very large and unstable updates to the weights and prevent the network from learning from the training data. This problem can occur in deep networks or recurrent neural networks.

To some extent, the exploding gradient problem can be mitigated by gradient clipping (thresholding the values of the gradients before performing a gradient descent step). Gradient clipping is a technique that limits the magnitude of the gradients during training. Another method to fix exploding gradients is weight regularization.

Answer_12:

The vanishing gradient problem is a problem that occurs during the training of deep neural networks, where the gradients that are used to update the network become extremely small or “vanish” as they are back-propagated from the output layers to the earlier layers. This makes it hard to learn and tune the parameters of the earlier layers in the network.

The vanishing gradient problem can cause the neural network to stop learning altogether or learn very slowly. This is because the gradients become so small that they do not provide enough information for the network to update its weights effectively. The problem can be mitigated by using activation functions such as ReLU (rectified linear unit) which can help prevent vanishing gradients.

Answer_13:

Regularization is a technique used to prevent overfitting in neural networks. Overfitting occurs when the model learns the training data too well and is unable to generalize to new data. Regularization methods like weight decay provide an easy way to control overfitting for large neural network models.

Regularization is any modification made to a learning algorithm that is intended to reduce its generalization error but not its training error. Regularization methods are so widely used to reduce overfitting that the term “regularization” may be used for any method that improves the generalization error of a neural network model. A modern recommendation for regularization is to use early stopping with dropout and a weight constraint.

Answer_14:

Normalization is a technique used to standardize the inputs of a neural network by adjusting and scaling the input values to have zero mean and unit variance. Normalization can help in improving the performance of neural networks by ensuring that the input and output data fall within a specific range.

Normalization can help training of our neural networks as the different features are on a similar scale, which helps to stabilize the gradient descent step, allowing us to use larger learning rates or help models converge faster for a given learning rate. Normalization can also reduce internal covariate shift which is the change in the distribution of network activations due to the change in network parameters during training.

Answer_15:

The most common activation functions used in a neural network are the sigmoid function, the Tanh function, and the ReLU function. The ReLU function is the most popular one, and it returns the input directly if it is positive, and 0 otherwise. The Rectifier Function is probably the most popular activation function in the world of neural networks. It is heavily used to solve all kinds of problems out there and for a good reason. This function is most biologically plausible of all functions described so far, and the most efficient function when it comes to training neural networks.

Answer_16:

Batch normalization is a technique that normalizes the inputs to a layer in a neural network to make training faster and more stable. It can be used with most network types, such as Multilayer Perceptrons, Convolutional Neural Networks and Recurrent Neural Networks. The benefit of batch normalization is that it reduces the internal covariate shift, which is the change in the distribution of the input vectors over time. This is important because the internal covariate shift can slow down training and cause the neural network to converge to a suboptimal solution.

This approach leads to faster learning rates since normalization ensures there’s no activation value that’s too high or too low, as well as allowing each layer to learn independently of the others. Normalizing inputs reduces the “dropout” rate, or data lost between processing layers.

Answer_17:

Weight initialization is a significant step employed before training any neural network. The weights of a network are initialized and then adjusted repeatedly while training the network. This is done till the loss converges to a minimum value and an ideal weight matrix is obtained. A proper initialization of the weights in a neural network is critical to its convergence.

There are several weight initialization strategies for neural networks. Some of them are random initialization, Xavier initialization, He initialization, and LeCun initialization. These strategies are used to initialize the weights of the neural network in such a way that it can converge faster and more accurately.

Answer_18:

Momentum is an optimization algorithm used in neural networks to speed up the training process. It is an adaptive optimization algorithm that uses exponentially weighted averages of gradients over previous iterations to stabilize the convergence, resulting in quicker optimization. The Momentum-based Gradient Optimizer has several advantages over the basic Gradient Descent algorithm, including faster convergence, improved stability, and the ability to overcome local minima. It is widely used in deep learning applications and is an important optimization technique for training deep neural networks.

Answer_19:

L1 and L2 regularization are two of the most common ways to reduce overfitting in deep neural networks. L1 regularization is performing a linear transformation on the weights of your neural network. L2 regularization is adding a squared cost function to your loss function. This cost function penalizes the sum of the absolute values of weights. The main intuitive difference between the L1 and L2 regularization is that L1 regularization tries to estimate the median of the data while the L2 regularization tries to estimate the mean of the data to avoid overfitting.

L2 regularization is also known as weight decay as it forces the weights to decay towards zero (but not exactly zero. In L2, we have:

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Here, lambda is the regularization parameter. It is the hyperparameter whose value is optimized for better results. In L1, we have:

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Answer_20:

Early stopping is a form of regularization used to avoid overfitting in neural networks by halting training when the error on the validation set starts to increase. It is one of the simplest and most intuitive regularization techniques. When training a neural network, we typically split the data into a training set and a validation set. Early stopping involves stopping the training of the neural network at an earlier epoch; hence the name early stopping. As you train the neural network over many epochs, the training error decreases. However, if you continue training for too long, the validation error will start to increase as the model starts to overfit. Early stopping is a useful technique for avoiding overfitting in neural networks.

Answer_21:

Dropout is a regularization technique for neural network models that randomly drops out some of the neurons in a layer during training.

Dropout prevents overfitting by forcing the network to learn redundant representations of the input data and approximating training a large number of neural networks with different architectures in parallel.

Dropout is a simple and computationally cheap way to regularize a deep neural network.

Dropout regularization is a technique to prevent neural networks from overfitting.

Dropout works by randomly disabling neurons and their corresponding connections. This prevents the network from relying too much on single neurons and forces all neurons to learn to generalize better.

Answer_22:

The learning rate is a configurable hyperparameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0. It controls how quickly the model is adapted to the problem and how much of the error value has to be back propagated to the weights in the network. The learning rate determines the size of the step or the magnitude of change to the model weights during the backpropagation process. The learning rate affects the convergence and the speed of the training. Smaller learning rates require more training epochs, whereas larger learning rates result in rapid changes and require fewer training epochs. The learning rate also controls the amount of apportioned error that the weights of the model are updated with each time they are updated.

In summary, it is important to choose an appropriate learning rate for your neural network as it can have a substantial influence on its performance.

Answer_23:

Training deep neural networks is a challenging and time-consuming process that involves finding the optimal weights to map inputs to outputs. Some of the major challenges include overfitting, exploding gradient, class imbalance, non-convex error surface, local minima, and flat spots. Optimization in general is an extremely difficult task.

Answer_24:

A convolutional neural network (CNN) is a type of neural network that is designed to process data with a grid-like topology, such as images. The main difference between a CNN and a regular neural network is that CNNs use convolutions to handle the math behind the scenes. A convolution is used instead of matrix multiplication in at least one layer of the CNN. Convolutions take two functions and return a function. CNNs work by applying filters to your input data.

Answer_25:

Pooling layers are used in convolutional neural networks to reduce the spatial dimensions of the output volume. The purpose of pooling is to reduce the spatial size of the input while retaining important information. Pooling layers work by sliding a window over the input and taking the maximum value within that window (max pooling) or taking the average value within that window (average pooling). This reduces the number of parameters in the model and helps to prevent overfitting.

Answer_26:

A recurrent neural network (RNN) is a type of neural network that is designed to recognize patterns in sequences of data, such as text, speech, or time series data. RNNs are used in a variety of applications, including image captioning, time series prediction, natural language processing (NLP), machine translation, speech recognition, generating image descriptions, video tagging, text summarization and call center analysis.

Answer_27:

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) that is designed to overcome the vanishing gradient problem present in traditional RNNs. LSTM networks can capture long-term dependencies in sequential data, which is difficult for traditional neural networks due to the vanishing gradient problem. LSTM cells can selectively remember or forget information using input gates, output gates, and forget gates.

LSTM has several benefits that make it a powerful tool for modeling sequential data:

- Ability to learn long-term dependencies
- Selective memory retention

LSTMs are much better at handling long-term dependencies than traditional RNNs due to their ability to remember information for extended periods of time. They are also much less susceptible to the vanishing gradient problem.

Answer_28:

A generative adversarial network (GAN) is a class of machine learning frameworks and a prominent framework for approaching generative AI. GANs are generative models that create new data instances that resemble your training data. The concept was initially developed by Ian Goodfellow and his colleagues in June 2014.

The GAN architecture consists of two neural networks competing against each other in a zero-sum game framework. The goal of GANs is to generate new, synthetic data that resembles some known data distribution.

Answer_29:

An autoencoder neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs. Autoencoders are neural networks that are composed of multiple layers. The defining aspect of an autoencoder is that the input layers contain exactly as much information as the output layer. The reason that the input layer and output layer have the exact same number of units is that an autoencoder aims to replicate the input data.

Autoencoders are used for a variety of tasks such as dimensionality reduction, feature extraction, and data compression. They can also be used for image denoising and image generation.

Answer_30:

A self-organizing map (SOM) is a type of artificial neural network that uses unsupervised learning to build a two-dimensional map of a problem space. SOMs are a data visualization technique invented by Professor Teuvo Kohonen which reduce the dimensions of data through the use of self-organizing neural networks.

SOMs are used for data visualization and clustering. They can be used for image recognition, speech recognition, and natural language processing.

Answer_31:

Neural networks can be used for regression tasks by using a loss function that is appropriate for regression. The output of the neural network is a continuous value rather than a discrete label representing a category.

Neural networks are flexible and can be used for both classification and regression. They can be applied to regression problems to establish a relationship between a dependent variable and one or more independent variables.

Answer_32:

Training neural networks with large datasets can be challenging due to the following reasons:

- ***Overfitting:*** Overfitting occurs when the model is too complex and starts to memorize the training data instead of learning the underlying patterns. This can be addressed by using regularization techniques such as dropout.

- ***Computation time:*** Training large neural networks can be computationally expensive and time-consuming. This can be addressed by using parallel computing techniques such as distributed training.

- ***Vanishing gradients:*** Vanishing gradients occur when the gradients become too small during backpropagation. This can be addressed by using activation functions such as ReLU.

- ***Memory constraints:*** Large datasets require a lot of memory to store and process. This can be addressed by using techniques such as mini-batch gradient descent.

Answer_33:

Transfer learning is a technique in which a pre-trained model is used as the starting point for a new model. The pre-trained model is trained on a large dataset and has learned to recognize general features that are useful for many tasks. The new model is then trained on a smaller dataset that is specific to the task at hand.

Transfer learning has several benefits such as:

- Saving time and resources
- Improving performance and robustness
- Enabling new applications and innovations
- Reducing training time
- Improving neural network performance (in most circumstances)
- Absence of a large amount of data
- Higher initial skill on the source model
- Steeper rate of improvement of skill during training of the source model
- Better converged skill of the trained model
- Reusing a model trained on a related predictive modeling problem
- Accelerating the training of neural networks as either a weight initialization scheme or feature extraction method.

Answer_34:

Neural networks can be used for anomaly detection in time series and image datasets.
Examples of neural network-based anomaly detection methods include:

- Time series segmentation approach based on convolutional neural networks (CNN)
- Supervised and semi-supervised detection of out-of-distribution samples in image datasets
- One-class neural network (OC-NN) model to detect anomalies in complex data sets

The one-class neural network (OC-NN) model combines the ability of deep networks to extract a progressively rich representation of data with the one-class objective of creating a tight envelope around normal data

Answer_35:

Neural networks are often criticized for their opaque nature, making it challenging to interpret the inner workings of neural network models. The original goal of proposing the neural network model was to improve the understanding of complex human brains using a mathematical expression approach. However, recent deep learning techniques continue to lose the interpretations of its functional process by being treated mostly as a black-box approximator.

There is a popular claim about the interpretability of machine learning models: Simple statistical models like logistic regression yield interpretable models. Neural networks, on the other hand, are black boxes. However, there are ways in which both linear models and neural networks are interpretable. They are both differentiable from top to bottom. Thus it is possible to trace the contribution of each input to the output (for a given example) and to show which inputs will move the output most if slightly increased or decreased.

We can certainly use model-agnostic methods, such as local models or partial dependence plots, but there are two reasons why it makes sense to consider interpretation methods developed specifically for neural networks: First, neural networks learn features and concepts in their hidden layers and we need special tools to uncover them. Second, interpretation methods developed specifically for neural networks can be more efficient than model-agnostic methods.

Answer_36:

Deep learning has several advantages over traditional machine learning methods, some of the main ones include:

- ***Automatic feature learning:*** Deep learning algorithms can automatically learn features from the data, which means that they don't require the features to be hand-engineered.
- ***Better performance:*** Deep learning algorithms can achieve higher accuracy than traditional machine learning algorithms.
- ***Scalability:*** Deep learning algorithms can scale with data, whereas traditional machine learning algorithms cannot.

However, deep learning also has some disadvantages compared to traditional machine learning algorithms:

- ***Requires more data:*** Deep neural networks require much more data than a traditional algorithm to ensure it trains accurately.
- ***Computationally expensive:*** Deep neural networks are computationally expensive and require powerful hardware to train.
- ***Difficult to interpret:*** Deep neural networks are often criticized for their opaque nature, making it challenging to interpret the inner workings of neural network models.

Answer_37:

Ensemble learning is a learning paradigm where multiple neural networks are jointly used to solve a problem. Ensemble learning combines the predictions from multiple neural network models to reduce the variance of predictions and reduce generalization error. Techniques for ensemble learning can be grouped by the element that is varied, such as training data, the model, and how predictions are combined.

A successful approach to reducing the variance of neural network models is to train multiple models instead of a single model and to combine the predictions from these models. This is called ensemble learning and not only reduces the variance of predictions but also can result in predictions that are better than any single model.

Answer_38:

Neural networks can be used for natural language processing (NLP) tasks such as sentiment analysis, language translation, and text classification. There are several types of neural networks that can be used for NLP tasks such as multilayer perceptron (MLP), convolutional neural network (CNN), recursive neural network (RNN), recurrent neural network (RNN), long short-term memory (LSTM), sequence-to-sequence models, and shallow neural networks.

Answer_39:

Self-supervised learning is a machine learning paradigm that processes unlabeled data to obtain useful representations that can help with downstream learning tasks. Self-supervised learning is particularly suitable for speech recognition. Self-supervised learning has achieved promising performance on natural language and image learning tasks. When labeled samples are limited, self-supervised learning (SSL) is emerging as a new paradigm for making use of large amounts of unlabeled samples.

Answer_40:

When trained on imbalanced datasets, neural networks tend to be biased towards the majority class, and they often fail to detect the minority class. Neural networks minimize a loss function during training, which measures the difference between the predicted output and the true output. Working with imbalanced datasets in neural networks can be challenging, but there are several techniques available to address this problem. These techniques include oversampling the minority class, undersampling the majority class, class weighting, cost-sensitive learning, and ensemble methods.

Answer_41:

Adversarial attacks are a method of making small modifications to the objects in such a way that the machine learning model begins to misclassify them. Neural networks are known to be vulnerable to such attacks. Adversarial examples refer to specially crafted inputs whose purpose is to force the neural network to misclassify them.

There are several techniques available to mitigate adversarial attacks on neural networks. These techniques include defensive distillation, adversarial training, input preprocessing, and model ensembling.

Answer_42:

The bias-variance tradeoff is a well-known concept in machine learning that refers to the trade-off between model complexity and generalization performance. Under the classical teachings of statistical learning, if you increase model complexity further, generalization error tends to increase (the typical U-shaped test error curve). However, recent work calls this into question for neural networks and other over-parameterized models, for which it is often observed that larger models generalize better.

Answer_43:

Neural networks are a powerful tool for modeling complex patterns in data, but they are susceptible to overfitting when there is missing data in the training set. One way to deal with missing data is to use a technique called imputation, which replaces missing values with plausible estimates. Another approach is to use a technique called dropout, which randomly drops neurons from the network during training. While neural networks require all inputs by default, there are several techniques that can be used to effectively handle missing data, such as imputation, masking, and specialized networks designed to handle missing data.

Answer_44:

SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are two popular model-agnostic, local explanation approaches designed to explain any given black-box classifier. These methods explain individual predictions of any classifier in an interpretable and faithful manner, by learning an interpretable model (e.g., linear model) locally around each prediction. SHAP values are model-agnostic, meaning they can be used to interpret any machine learning model, including neural networks. SHAP value is a real breakthrough tool in machine learning interpretation. SHAP value can work on both regression and classification problems. Also works on different kinds of machine learning models like logistic regression, SVM, tree-based models and deep learning models like neural networks.

Answer_45:

Deploying neural networks directly to edge devices comes with many advantages compared to traditional cloud deployments. It eliminates communication which can reduce latency and reliance on the network connection; since the data never leaves the device, edge-inference helps with maintaining user privacy; and since the amount of cloud resources is drastically reduced, edge-inference can also reduce ongoing costs. To deploy DNNs on edge devices, we need to reduce the size of DNNs, i.e., we target a better trade-off between resource consumption and model accuracy. There are execution frameworks that accelerate deep neural network inference through device-edge synergy.

Answer_46:

Scaling neural network training on distributed systems is challenging. One of the main challenges of distributed training is maintaining acceptable training time performance. Training deep learning neural networks is very challenging. The best general algorithm known for solving this problem is stochastic gradient descent, where model weights are updated each iteration using the backpropagation of error algorithm. Problems of this magnitude are common and thus researching parallel network optimization on distributed and parallel systems is highly important. There are many different tactics for scaling up neural network training. Early work in training large distributed neural networks focused on schemes for partitioning networks over multiple cores, often referred to as model parallelism.

Answer_47:

The use of neural networks in decision-making systems raises a number of ethical questions about identity, privacy, responsibility, and justice. One of the interesting things about neural networks is that they effectively merge a computer program with the data that is given to it. This has many benefits, but it also risks biasing the entire system in unexpected and potentially detrimental ways. Companies have to think seriously about the ethical dimensions of what they’re doing and we, as democratic citizens, have to educate ourselves about tech and its social and ethical implications — not only to decide what the regulations should be but also to decide what role we want big tech and social media to play in our lives.

Answer_48:

Reinforcement learning is a framework that helps software agents learn how to reach their goals. It is a learning method where a software agent interacts with an unknown environment, selects actions, and progressively discovers the environment dynamics. Deep reinforcement learning combines this framework with artificial neural networks to map states and actions to the rewards they lead to. In deep reinforcement learning, a neural network is used in reinforcement learning to represent policies or value functions. Reinforcement learning has been applied in various fields such as robotics, game playing, finance, healthcare and more.

Answer_49:

Batch size has a significant impact on the speed, accuracy, and stability of neural network training. A larger batch size means that more data can be processed in parallel, which can speed up the training and reduce the memory requirements. However, larger batch sizes can lead to slower training loss decreases, higher minimum validation loss, and more epochs to converge to the minimum validation loss. There is a tension between batch size and the speed and stability of the learning process. Research has found that classification accuracy increases with batch size.

Answer_50:

Neural networks have some limitations that are worth noting. One of the best-known disadvantages of neural networks is their “black box” nature. Simply put, you don’t know how or why your NN came up with a certain output. Another issue is that neural networks are currently being deployed everywhere; it’s become the hammer to which everything is a nail. There are also mathematical paradoxes that demonstrate the limits of AI.

As for areas for future research, there are many. Some of the most promising areas include developing more efficient algorithms for training neural networks, improving the interpretability of neural networks, and developing new architectures that can handle more complex data structures.