LP5 Sample Question Bank

What is Keras?
A: Keras is a Python-based deep learning library that is compiled to run over a number of platforms, including TensorFlow and Theano. It is a high-level API that simplifies the process of building and training neural networks.
Q: What are the main advantages of using Keras?
A: Keras offers several advantages, including:

- **User-friendly API:**
  Keras provides a clean and intuitive API for building and training neural networks.
- **Simplified model building:**
  Keras allows users to define and train models quickly and easily, with a focus on ease of use and readability.
- **Support for various backends:**
  Keras can run on different backends, including TensorFlow and Theano, allowing users to choose the one that best suits their needs.
- **Ease of debugging:**
  Keras is designed for human beings, making it easier to debug and troubleshoot.
- **Maintainability and deployability:**
  Keras code is often more readable and maintainable, making it easier to deploy models in production.

Q: How nstall Keras?
A: Keras can be installed using pip: pip install keras.

Q.Explain the term regularization.
Answer: Regularization is a method that makes slight modifications to the learning algorithm such that the model generalizes better. This in turn improves the model's performance on the unseen data as well.

Q.Name some of the regularization techniques.
Answer:
The techniques are as follows:

1. L2 and L1 Regularization
2. Dropout
3. Early Stopping
4. Data Augmentation

Q. What are the key parameters to adjust in a linear regression model?

- **Input shape:** This specifies the number of input features.
- **Activation function:** For linear regression, it should be set to "linear".
- **Optimizer:** Choose an optimizer like "adam" or "sgd".
- **Loss function:** "mean_squared_error" is a common choice for regression.

    - **Metrics:** Use metrics like "mse" or "mae" to evaluate performance.
    - **Epochs:** The number of passes through the entire dataset.
    - **Batch size:** The number of samples processed in each iteration.

Q.What is a Convolutional Neural Network (CNN)?

- **Answer:** A CNN is a type of neural network that is specially designed for processing grid-like data, such as images. It uses convolutional layers to extract features, pooling layers to reduce dimensionality, and fully connected layers for classification.

Q.What are convolutional layers?

- **Answer:** Convolutional layers apply a filter (also called a kernel) across the input image, extracting features such as edges, textures, and shapes. These features are then used to build more complex representations in subsequent layers.

Q. What are pooling layers?

Q: What are the different types of models in Keras?

A: Keras provides two main types of models:

- **Sequential model:**
  This is a linear stack of layers, where each layer processes the output of the previous layer.
- **Functional model:**
  This allows for more complex model architectures, with multiple input and output paths.

Q: What are activation functions?

A: Activation functions introduce non-linearity into neural networks. They determine how the output of a layer is transformed based on its input.

Q: What are some common activation functions?

A: Some common activation functions include ReLU (Rectified Linear Unit), sigmoid, and tanh.

Q: What is the purpose of the compile() method in Keras?

A: The compile() method configures the model for training by defining the loss function, optimizer, and metrics.

Q: What is the purpose of the fit() method in Keras?

A: The fit() method trains the model by feeding it the training data and hyperparameters.

Q: What is the purpose of the predict() method in Keras?

A: The predict() method uses the trained model to make predictions on new, unseen data.

Question 1: What are Recurrent Neural Networks (RNNs) and how do they differ from Feedforward Neural Networks?

Answer: RNNs are designed to process sequential data, unlike Feedforward networks that process static data. RNNs have a "memory" (hidden state) that remembers past inputs, allowing them to learn from the order of information. Feedforward networks don't have this memory and process each input independently.

Question 2: Explain the concept of time steps in the context of RNNs.

Answer: In RNNs, time steps represent the sequential flow of data. Each input, hidden state, and output is associated with a specific time point, allowing the network to process data sequentially.

Question 3: What are some common RNN architectures?

Answer: Common RNN architectures include SimpleRNN, LSTM, and GRU. SimpleRNN is a basic RNN, while LSTM and GRU are designed to address the vanishing gradient problem in vanilla RNNs.

Q Describe how *Recurrent Neural Networks* are different and how to implement one in *Keras*?
A Recurrent Neural Network (RNN) processes sequential data by maintaining a hidden state that encapsulates the network's knowledge up to a certain point in the sequence.
For instance, if our input sequence is (x1,x2,x3,…,xT), the RNN updates its hidden state at each time step t using the input at that time step xt and the previous hidden state ht−1.
Mathematically, this is represented as:
ht=RNN(xt,ht−1)

Q What is Long Short-Term Memory Network for RNN
The Long Short-Term Memory network, or LSTM network, is a recurrent neural network trained using Backpropagation Through Time that overcomes the vanishing gradient problem.
As such, it can be used to create large recurrent networks that, in turn, can be used to address difficult sequence problems in machine learning and achieve state-of-the-art results.
Instead of neurons, LSTM networks have memory blocks connected through layers.
A block has components that make it smarter than a classical neuron and a memory for recent sequences. A block contains gates that manage the block's state and output. A block operates upon an input sequence, and each gate within a block uses the sigmoid activation units to control whether it is triggered or not, making the change of state and addition of information flowing through the block conditional.
There are three types of gates within a unit:

- Forget Gate: conditionally decides what information to throw away from the block
- Input Gate: conditionally decides which values from the input to update the memory state
- Output Gate: conditionally decides what to output based on input and the memory of the block

Each unit is like a mini-state machine where the gates of the units have weights that are learned during the training procedure.
You can see how you may achieve sophisticated learning and memory from a layer of LSTMs, and it is not hard to imagine how higher-order abstractions may be layered with multiple such layers.

Q Define the Model Architecture for RNN with LSTM:

- Initialize a Sequential model.
- Add an LSTM layer using model.add(LSTM()).
    - Specify the number of units for the LSTM layer. This determines the complexity of the LSTM cell.
    - Optionally, use return_sequences=True when stacking multiple LSTM layers so that the output of one LSTM layer is used as input for the next.
- Add a Dense layer for the final output.
    - Adjust the number of units in the Dense layer depending on the number of output classes or dimensions.

ALL THE BEST

Nilam S. Patil
Subject Teacher