Viva Questions on TensorFlow Neural Network for Classification

Basic Understanding

1. **What is the purpose of this code?**
   - The code builds, trains, and evaluates a neural network model using TensorFlow to predict customer churn based on a dataset.
2. **What is TensorFlow?**
   - TensorFlow is an open-source machine learning library developed by Google for building and training machine learning models.
3. **What is the role of `Sequential` in this code?**
   - `Sequential` is a linear stack of layers in TensorFlow's Keras API, where you can add layers one by one to build a model.
4. **Why do we use `Dense` layers in the model?**
   - `Dense` layers are fully connected layers where each neuron is connected to every neuron in the previous layer. They are commonly used in feedforward neural networks.
5. **What does the `LabelEncoder` do in the preprocessing step?**
   - `LabelEncoder` converts categorical variables into numerical form so that they can be processed by the machine learning model.

Data Preparation

6. **Why do we drop the 'Surname' column?**
   - The 'Surname' column is not relevant for prediction as it's just an identifier with no predictive value.
7. **What does `train_test_split` do?**
   - It splits the dataset into training and testing sets, where the model is trained on the training set and evaluated on the testing set.
8. **Why do we use the `fit_transform` method on categorical columns like 'Geography' and 'Gender'?**
   - `fit_transform` encodes the categorical columns into numerical values to be used by the model for training.

Model Architecture

9. **What does `input_dim=12` specify in the first `Dense` layer?**
   - It specifies the number of input features (in this case, 12), corresponding to the number of columns in the feature set.
10. **Why do we use `ReLU` as an activation function in the hidden layers?**
    - ReLU (Rectified Linear Unit) introduces non-linearity into the model, helping it learn complex patterns. It also helps to avoid the vanishing gradient problem.

11. **What is the purpose of using the `sigmoid` activation function in the output layer?**
    - The `sigmoid` function is used in binary classification tasks to output a probability score between 0 and 1.
12. **Explain the loss function used here (`binary_crossentropy`).**
    - `binary_crossentropy` is a loss function used in binary classification problems. It calculates the difference between the predicted probabilities and the actual labels.
13. **What are other types of loss functions used in neural networks?**
    - **`sparse_categorical_crossentropy`:** Used for multi-class classification where classes are mutually exclusive.
    - **`mean_squared_error (MSE)`:** Used for regression tasks to minimize the difference between predicted and actual values.
    - **`categorical_crossentropy`:** Used for multi-class classification tasks with one-hot encoded labels.
    - **`hinge`:** Often used for "maximum-margin" classification, most notably for support vector machines.
14. **Why is `Adam` optimizer chosen?**
    - `Adam` is a popular optimizer that combines the advantages of two other extensions of stochastic gradient descent: AdaGrad and RMSProp. It's efficient and works well in practice.
15. **What are some other types of optimizers in TensorFlow?**
    - **`SGD (Stochastic Gradient Descent)`:** Basic optimizer that updates the weights using the gradient of the loss function.
    - **`RMSprop`:** Adaptive learning rate method that maintains a moving average of the squared gradient.
    - **`Adagrad`:** Adapts the learning rate to the parameters, performing smaller updates for frequently occurring features.
    - **`Adadelta`:** An extension of Adagrad that seeks to reduce its aggressive, monotonically decreasing learning rate.
    - **`Nadam`:** A combination of Adam and Nesterov momentum.
16. **How does one define a model architecture in TensorFlow?**
    - Model architecture is defined by stacking layers in a `Sequential` model or using the Functional API for more complex architectures. You specify the type of layers (e.g., Dense, Conv2D, LSTM) and their configurations (e.g., number of units, activation functions).

Model Training

17. **What does the `epochs=10` argument mean in the `fit` method?**
    - It specifies that the model will go through the entire training dataset 10 times.
18. **What does the `model.summary()` function do?**
    - It provides a summary of the model architecture, including the number of layers, output shape, and the number of parameters in each layer.
19. **Why do we use the `predict` method on the test data?**

- The `predict` method generates predictions for the test data using the trained model.

20. **Why do we use `np.argmax(y_pred, axis=1)`?**
    - `np.argmax` is incorrectly used here because the model outputs a single value per instance due to the sigmoid activation. This would be used in multi-class classification, but for binary, we should threshold predictions at 0.5 instead.
21. **What is the purpose of setting a random seed in the code (`random_state=108`)?**
    - Setting a random seed ensures reproducibility, meaning that the same data split will be used every time the code is run.

Model Evaluation and Metrics

22. **Explain what a confusion matrix is.**
    - A confusion matrix is a table that summarizes the performance of a classification model by showing the number of true positives, true negatives, false positives, and false negatives.
23. **What information does the classification report provide?**
    - The classification report provides precision, recall, F1-score, and support for each class, offering a detailed analysis of model performance.
24. **What is precision?**
    - Precision is the ratio of correctly predicted positive observations to the total predicted positives. It answers the question: *What proportion of positive identifications was actually correct?*
25. **What is recall?**
    - Recall is the ratio of correctly predicted positive observations to all observations in the actual class. It answers the question: *What proportion of actual positives was identified correctly?*
26. **What is F1-score?**
    - The F1-score is the weighted average of precision and recall. It seeks a balance between precision and recall and is especially useful when the classes are imbalanced.
27. **What is support in the classification report?**
    - Support refers to the number of actual occurrences of the class in the dataset. It helps to understand how the metrics are distributed among different classes.
28. **What is accuracy, and how does it differ from precision and recall?**
    - Accuracy is the proportion of correctly predicted observations to the total observations. Unlike precision and recall, it gives a general measure of performance across all classes, which can be misleading if the classes are imbalanced.
29. **Why do we plot a heatmap of the confusion matrix?**
    - A heatmap provides a visual representation of the confusion matrix, making it easier to interpret the results.

30. **How do different metrics like accuracy, precision, recall, and F1-score complement each other?**
    - Accuracy gives a general measure of performance, while precision and recall provide insights into the model's ability to correctly identify positive instances. F1-score balances precision and recall, making it more suitable when dealing with imbalanced datasets.
31. **What could you do if the model's performance is not satisfactory?**
    - Consider tuning hyperparameters, adding more data, using different features, trying a different model architecture, or performing data augmentation.
32. **How would you handle an imbalanced dataset in this scenario?**
    - Techniques such as oversampling the minority class, undersampling the majority class, or using techniques like SMOTE (Synthetic Minority Over-sampling Technique) could be applied.
33. **What is overfitting and how can it be prevented in this model?**
    - Overfitting occurs when the model learns patterns specific to the training data, leading to poor generalization on unseen data. It can be prevented by using regularization techniques, dropout, or early stopping.
34. **Why is it important to split data into training and testing sets?**
    - Splitting data helps evaluate the model's ability to generalize to new, unseen data, ensuring that it doesn't just memorize the training data.
35. **Can this model be used for multi-class classification?**
    - Yes, by modifying the output layer and loss function, the model can be adapted for multi-class classification tasks.
36. **What other neural network architectures could be used for this task?**
    - Convolutional Neural Networks (CNNs) for image data, Recurrent Neural Networks (RNNs) for sequential data, or more complex architectures like LSTMs or GRUs could be used depending on the data type.
37. **How can you improve the model's accuracy?**
    - Techniques such as hyperparameter tuning, increasing model complexity, or using ensemble methods can be explored to improve accuracy.
38. **What are the challenges of using deep learning models like this one?**
    - Deep learning models require large amounts of data, significant computational resources, and careful tuning of hyperparameters. They can also be prone to overfitting and require careful evaluation.
39. **What is the role of dropout in preventing overfitting?**
    - Drop

out randomly deactivates a fraction of neurons during training, forcing the network to learn more robust features and preventing it from relying too heavily on any single neuron.

40. **Explain the concept of batch processing during model training.**

- Batch processing involves dividing the dataset into smaller batches, allowing the model to be trained on smaller portions of data at a time, which helps manage memory usage and can lead to faster convergence.