Training Day-91 Report:

Keras and **Sequential Composition**:

Definition of Keras
Keras is an open-source high-level neural network API written in Python. It is integrated with TensorFlow and provides an intuitive interface to build, train, and deploy deep learning models.
Key Features of Keras:

1. **User-Friendly:**
    - Designed for fast experimentation and ease of use.
2. **Modular Structure:**
    - Models are built by connecting modular components like layers, optimizers, and loss functions.
3. **Flexibility:**
    - Supports simple models using the Sequential API and complex models using the Functional or Subclassing APIs.
4. **Scalability:**
    - Runs seamlessly on CPUs, GPUs, and TPUs.
5. **Integration with TensorFlow:**
    - Keras is now part of TensorFlow (tf.keras), leveraging TensorFlow's power while retaining its simplicity.

Sequential Composition in Keras

Definition:
Sequential Composition in Keras refers to creating a model layer-by-layer in a linear stack. This approach is ideal for straightforward neural networks where each layer connects to the one before it.
Steps to Create a Sequential Model:

1. **Initialize the Model:**
    - Use tf.keras.Sequential() to define the model.
2. **Add Layers:**
    - Stack layers sequentially using the .add() method or within the Sequential constructor.
3. **Compile the Model:**
    - Specify the optimizer, loss function, and metrics for training.
4. **Train the Model:**
    - Use the model.fit() method to train with data.

Example of a Sequential Model:

```
import tensorflow as tf

# Create a Sequential model
model = tf.keras.Sequential([
   tf.keras.layers.Dense(128, activation='relu', input_shape=(784,)),
   tf.keras.layers.Dropout(0.2),  # Dropout layer to prevent overfitting
   tf.keras.layers.Dense(10, activation='softmax')  # Output layer for classification
])

# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Summary of the model
model.summary()
```

When to Use Sequential Composition:

- When the network is simple and has a single input and output.
- For linear architectures, where layers are stacked in a straightforward order.

Limitations of Sequential Composition

- Cannot handle complex models with:
    - Multiple inputs or outputs.
    - Shared layers.
    - Non-linear architectures.

For such cases, use the **Functional API** or **Subclassing API**.