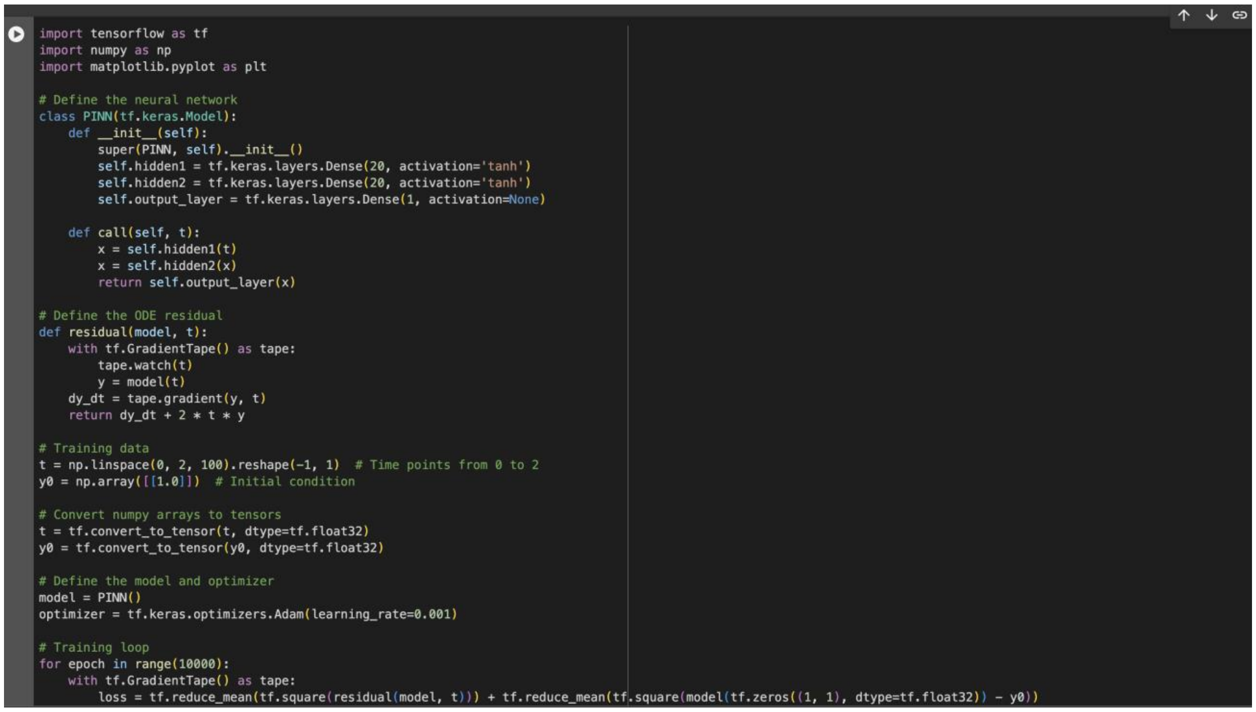
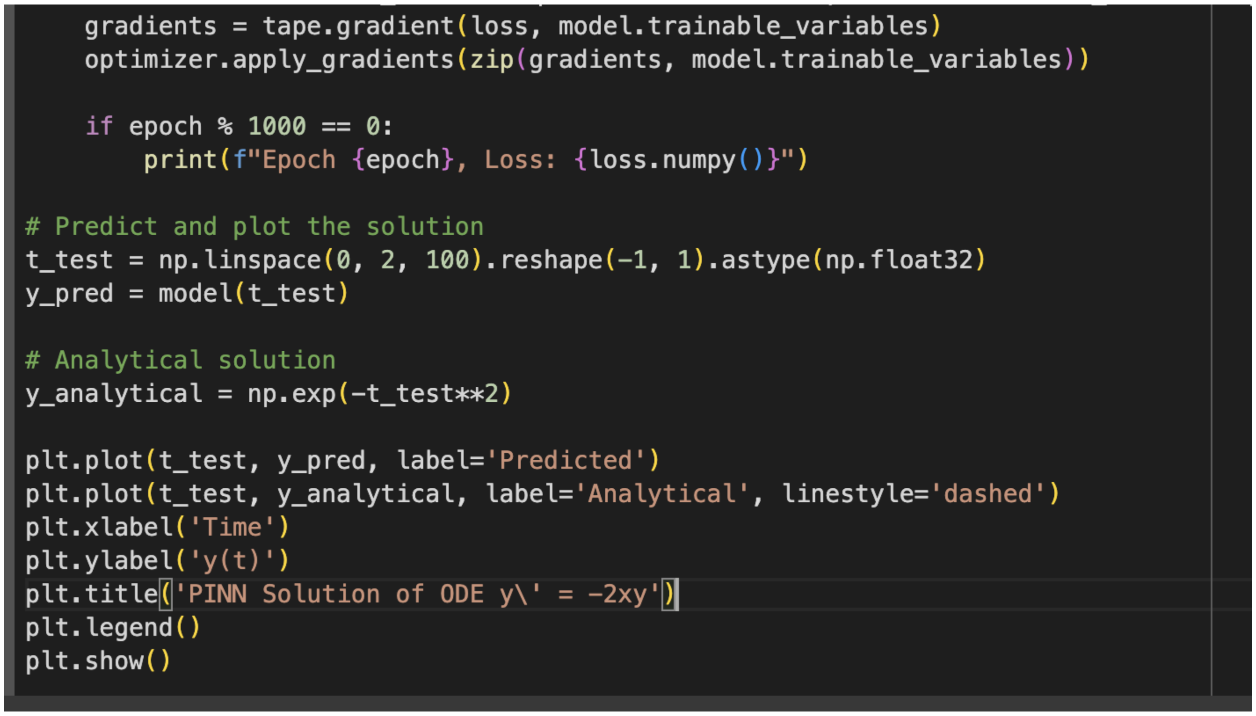
Akiti Sri kalyan Reddy

Solving differential equation using deep learning technique.

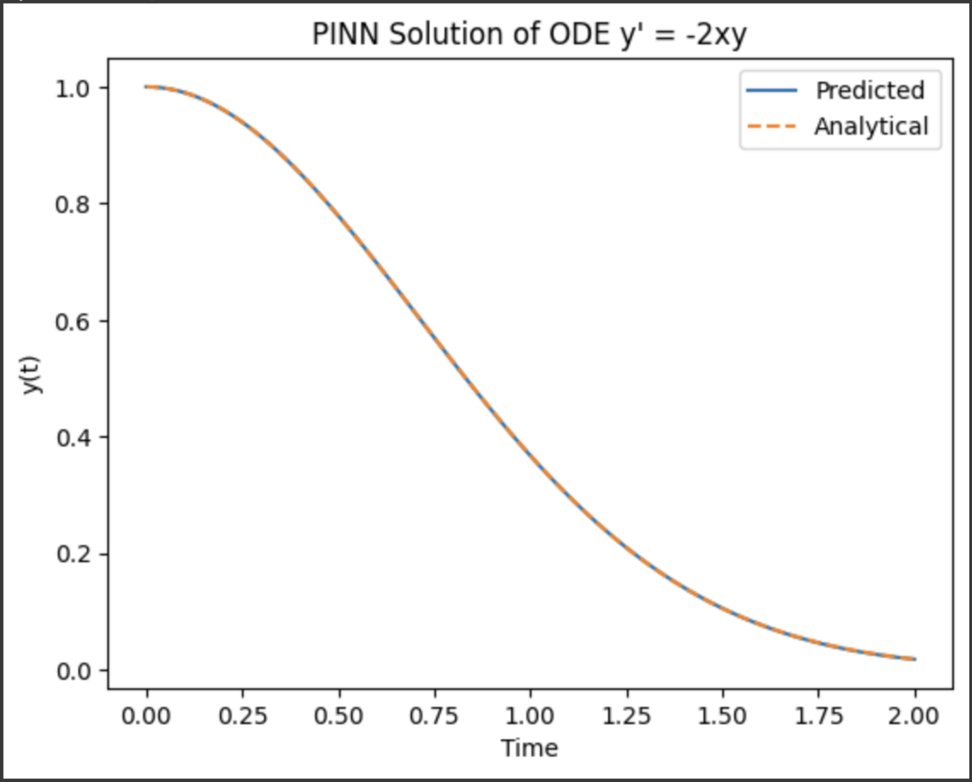


```python
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

# Define the neural network
class PINN(tf.keras.Model):
    def __init__(self):
        super(PINN, self).__init__()
        self.hidden1 = tf.keras.layers.Dense(20, activation='tanh')
        self.hidden2 = tf.keras.layers.Dense(20, activation='tanh')
        self.output_layer = tf.keras.layers.Dense(1, activation=None)

    def call(self, t):
        x = self.hidden1(t)
        x = self.hidden2(x)
        return self.output_layer(x)

# Define the ODE residual
def residual(model, t):
    with tf.GradientTape() as tape:
        tape.watch(t)
        y = model(t)
    dy_dt = tape.gradient(y, t)
    return dy_dt + 2 * t * y

# Training data
t = np.linspace(0, 2, 100).reshape(-1, 1)  # Time points from 0 to 2
y0 = np.array([[1.0]])  # Initial condition

# Convert numpy arrays to tensors
t = tf.convert_to_tensor(t, dtype=tf.float32)
y0 = tf.convert_to_tensor(y0, dtype=tf.float32)

# Define the model and optimizer
model = PINN()
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)

# Training loop
for epoch in range(10000):
    with tf.GradientTape() as tape:
        loss = tf.reduce_mean(tf.square(residual(model, t))) + tf.reduce_mean(tf.square(model(tf.zeros((1, 1), dtype=tf.float32)) - y0))
```



```python
    gradients = tape.gradient(loss, model.trainable_variables)
    optimizer.apply_gradients(zip(gradients, model.trainable_variables))

    if epoch % 1000 == 0:
        print(f"Epoch {epoch}, Loss: {loss.numpy()}")

# Predict and plot the solution
t_test = np.linspace(0, 2, 100).reshape(-1, 1).astype(np.float32)
y_pred = model(t_test)

# Analytical solution
y_analytical = np.exp(-t_test**2)

plt.plot(t_test, y_pred, label='Predicted')
plt.plot(t_test, y_analytical, label='Analytical', linestyle='dashed')
plt.xlabel('Time')
plt.ylabel('y(t)')
plt.title('PINN Solution of ODE y\' = -2xy')
plt.legend()
plt.show()
```

```
Epoch 0, Loss: 1.1709338426589966
Epoch 1000, Loss: 0.0003786842280533165
Epoch 2000, Loss: 0.0001355494314339012
Epoch 3000, Loss: 4.677416291087866e-05
Epoch 4000, Loss: 3.283169644419104e-05
Epoch 5000, Loss: 2.6944546334561892e-05
Epoch 6000, Loss: 3.9500264392700046e-05
Epoch 7000, Loss: 9.3334965640679e-05
Epoch 8000, Loss: 1.0818866030604113e-05
Epoch 9000, Loss: 7.177952284109779e-06
```


PINN Solution of ODE y' = -2xy
Predicted
Analytical
y(t)
Time