Transformer Ablation Experiment on Google Cloud TPU (TAETPU)

This repository contains a framework for conducting Transformer model ablation experiments on Google Cloud TPUs. It provides the infrastructure to set up, run, and analyze experiments that examine the impact of various Transformer architecture components.

Project Structure

```
.
├── .gitattributes                # Git attributes configuration
├── .gitignore                    # Git ignore configuration
├── README.md                     # Project documentation (this file)
├── config/                       # Configuration and credential files
│   ├── .env                      # Environment variables and configuration
│   ├── requirements.txt          # Python dependencies for the project
│   └── infra-tempo-####          # Service account key
├── infrastructure/               # Infrastructure setup and management
│   ├── setup/                    # Scripts for setting up the environment
│   │   ├── check_zones.sh        # Script to find available TPU zones
│   │   ├── setup_bucket.sh       # Script to create GCS bucket
│   │   ├── setup_image.sh        # Script to build and push Docker image to
GCR
│   │   └── setup_tpu.sh          # Script to create TPU VM and pull Docker
image
│   ├── docker/                   # Docker configuration
│   │   ├── Dockerfile            # Docker image definition
│   │   ├── docker-compose.yml    # Docker Compose configuration
│   │   ├── entrypoint.sh         # Container entry point script
│   │   └── requirements.txt      # Python dependencies for Docker
│   ├── utils/                    # Shared utilities
│   │   ├── common.sh             # Common bash utilities and functions
│   │   ├── monitors/             # Monitoring utilities
│   │   ├── logging/              # Logging utilities
│   │   └── backend/              # Backend utilities
│   └── teardown/                 # Scripts for resource cleanup
│       ├── teardown_bucket.sh    # Script to delete GCS bucket
│       ├── teardown_image.sh     # Script to clean up Docker images
│       └── teardown_tpu.sh       # Script to delete TPU VM
└── tools/                        # Tools for operations and management
    ├── docker_ops/               # Docker operations
    │   ├── mgt/                  # Management scripts
    │   │   ├── mount.sh          # Script to mount files to TPU VM
    │   │   ├── run.sh            # Script to execute files on TPU VM
    │   │   └── scrap.sh          # Script to remove files from TPU VM
    │   └── src/                  # Source code for Docker operations
    └── gcs_ops/                  # GCS storage operations
```

```
        ├── data_ops.sh           # Unified data operations script
        └── downloads/            # Local dataset storage
```

0. Requirements

Before starting, ensure you have:

- Docker Desktop installed and running
- Google Cloud SDK installed and configured
- Python 3.11+ for local development
- Google Cloud account with billing enabled
- Service account with appropriate permissions
- Git for version control

Make all scripts executable:

```
chmod +x infrastructure/setup/*.sh
chmod +x infrastructure/teardown/*.sh
chmod +x infrastructure/utils/*.sh
chmod +x tools/gcs_ops/data_ops.sh
chmod +x tools/docker_ops/mgt/*.sh
```

1. Configuration

Create and configure your environment variables:

```
# Copy the template (don't edit the template directly)
cp config/.env.template config/.env

# Edit your .env file with your specific settings
nano config/.env  # or use your preferred editor
```

Your `.env` file should contain:

```
# Project Configuration
PROJECT_ID=your-project-id
TPU_REGION=europe-west4
TPU_ZONE=europe-west4-a
BUCKET_REGION=europe-west4
TPU_NAME=your-tpu-name
TPU_TYPE=v2-8
RUNTIME_VERSION=tpu-ubuntu2204-base

# Cloud Storage
BUCKET_NAME=your-bucket-name
```

```
BUCKET_DATRAIN=your-bucket-name/data
BUCKET_TENSORBOARD=your-bucket-name/tensorboard

# Service Account details
SERVICE_ACCOUNT_JSON=your-service-account.json
SERVICE_ACCOUNT_EMAIL=your-service-account@your-project.iam.gserviceaccount.com

# Dataset Configuration
# Format: DATASET_[KEY]_NAME = the dataset name/path on Hugging Face
DATASET_GUTENBERG_NAME="nbeerbower/gutenberg2-dpo"
DATASET_EMOTION_NAME="dair-ai/emotion"
```

2. Setup Process

2.1 Check for Available TPU Zones

Find a zone where your desired TPU type is available:

```
./infrastructure/setup/check_zones.sh
```

2.2 Set Up Google Cloud Storage Bucket

Create a bucket for storing experiment data, model checkpoints, and logs:

```
./infrastructure/setup/setup_bucket.sh
```

2.3 Build and Push the Docker Image

Build your Docker image and push it to Google Container Registry:

```
./infrastructure/setup/setup_image.sh
```

2.4 Set Up TPU VM and Pull Docker Image

Create the TPU VM, pull the Docker image, and start the container:

```
./infrastructure/setup/setup_tpu.sh
```

3. Development Workflow

3.1 Data Operations

The `data_ops.sh` script provides data management between your local machine and Google Cloud Storage:

```
# Show help information
./tools/gcs_ops/data_ops.sh --help

# Download datasets to local machine
./tools/gcs_ops/data_ops.sh download-local

# Upload local datasets to GCS bucket
./tools/gcs_ops/data_ops.sh upload --bucket-name your-bucket-name

# Download datasets from GCS bucket to TPU VM
./tools/gcs_ops/data_ops.sh download-gcs --bucket-name your-bucket-name

# List available datasets in GCS bucket
./tools/gcs_ops/data_ops.sh list --bucket-name your-bucket-name
```

3.2 Working with TPU VM

Mount and run files on the TPU VM:

```
# Mount files to TPU VM
./tools/docker_ops/mgt/mount.sh example.py

# Run a file on TPU VM
./tools/docker_ops/mgt/run.sh example.py

# Clean up files from TPU VM
./tools/docker_ops/mgt/scrap.sh --all
```

4. Teardown Resources

When you're done with your TPU resources:

```
# Delete the TPU VM
./infrastructure/teardown/teardown_tpu.sh

# Delete the Docker images (local and GCR)
./infrastructure/teardown/teardown_image.sh

# Delete the GCS bucket (will prompt for confirmation)
./infrastructure/teardown/teardown_bucket.sh
```

5. Additional Resources

- Google Cloud TPU Documentation
- PyTorch XLA Documentation
- TPU Performance Guide