**Scientific Project Description for Neural Network Implementation**

**Project Overview**This project involves building a neural network model designed to forecast variations in the Earth's surface magnetic field. The model integrates data from two distinct inputs (branches):

* **Branch 1 (Large-scale dynamics)**: Utilizes solar wind and global-scale geophysical data.
* **Branch 2 (Small-scale dynamics)**: Employs detailed ground-based magnetometer data processed with the Spherical Elementary Current Systems (SECS) method.

The goal is to predict magnetic field variations at the central point of a 21×21 SECS grid for forecast intervals ranging from 1 to 30 minutes ahead.

**Data Description and Organization**

**Target Data**

* Predictions will focus on three magnetic field components (Be, Bn, Bu) at the grid center.
* Target data is stored in CSV format target.csv. Where column one is the Date-Time information, column 2 is the Be, 3 is Bn, 4 is Bu.
* The data is organized by magnetic field component (Be, Bn, Bu) and located in:  
  /Users/akv020/Tensorflow/fennomag-net/source/model2024/data/target.csv

**Branch 1: Large-Scale Solar and Geophysical Data**

* **Purpose**: Captures broad interactions between solar and terrestrial systems for long-duration forecasting.
* **Data**: Stored in a CSV file at:  
  /Users/akv020/Tensorflow/fennomag-net/source/model2024/data/geodata.csv
* **Timeframe**: Data spans from January 1, 2024, to December 31, 2024, recorded every 15 minutes.
* **Data Features (columns)**:
  1. DateTime
  2. SME, SML, SMU (auroral electrojet currents)
  3. SYM\_D, SYM\_H (global ring current indicators)
  4. ASY\_D, ASY\_H (geomagnetic disturbance measures)
  5. Sunspot (solar activity proxy)
  6. f107 (solar radio flux)
  7. ap\_index (geomagnetic activity)
  8. Lyman (solar UV radiation)
  9. DOY\_cos/sin, TOD\_cos/sin (seasonal and daily encodings)
  10. SolarZenithAngle
  11. BE, BN, BU (Previous Mean Magnetic field observations – between t = 0 and t = -15 min)
  12. stdE, stdN, stdUthe standard deviation of over the previous 15 min observation period (between t = 0 and t = -15 min)
* **Data Preprocessing**:
  1. Chronologically split into training (70%), validation (15%), and testing (15%) datasets.
  2. 24-hour lookback windows (96 timesteps) created using a sliding window every 15 minutes are to be used.

**Branch 2: Small-Scale SECS-based Magnetometer Data**

* **Purpose**: Captures rapid ionospheric variations essential for short-term forecasting.
* **Data Format**: Stored as RGB images in one .npy array with dimension (n\_obs, W,H,C), where W = H = 21, C = 3 and n\_obs is the number of observations.
* **Location**:  
  /Users/akv020/Tensorflow/fennomag-net/source/model2024/data/secs\_data.npy

Alongside the timestap information in the .npy file:  
/Users/akv020/Tensorflow/fennomag-net/source/model2024/data/secs\_timestamps.npy

* **Data Dimensions**: (21×21×3×180) representing spatial dimensions, RGB channels for magnetic components, and 180 timesteps (3 hours at 1-minute intervals) are to be used.

**Batch Processing and Data Handling**

* **Batch Processing Strategy:** The model processes data in batches to efficiently handle the large dataset while maintaining memory efficiency.
* **Each batch contains:**
  + Branch 1: 96 timesteps of solar/geophysical data (15-minute intervals)
  + Branch 2: 180 timesteps of SECS image data (1-minute intervals)
  + Target: Corresponding magnetic field predictions for the next X minutes
* **Batch size is set to 64 samples to balance:**
  + Memory efficiency
  + Training stability
  + Computational performance
* **The batch generator handles:**
  + Proper alignment of Branch 1 and Branch 2 data
  + Temporal synchronization of different sampling rates
  + Validation of data availability within observation periods
* **Data is processed sequentially through the network:**
  + 1. Branch 1 data → LSTM processing
    2. Branch 2 data → Conv layers → LSTM processing
    3. Cross-attention fusion of both branches
    4. Final prediction generation
* **For efficient visualization and analysis:**
  + Single batches can be processed independently
  + Random days within a batch can be selected for visualization
  + Performance metrics (RMSE) are calculated per component and time period

**Methodology and Network Architecture**

**Model Structure**

The neural network follows a three-stage process:

* **Stage 1: Feature Extraction**
  + Branch 1 data processed via LSTM networks to generate embeddings (B1).
  + Branch 2 data processed via Conv layers then an LSTM to generate embeddings (B2).
* **Stage 2: Cross-Attention Fusion**
  + Combines embeddings using cross-attention mechanisms, allowing detailed querying between Branch 1 and Branch 2.
  + Produces a combined embedding (B\_att) to capture global-local interactions explicitly.
* **Stage 3: Decoder for Prediction Output**
  + Combined embedding is processed through fully connected layers.
  + Output: Predictions of magnetic field components at the grid center for horizons of e.g. 1, 5, 10, 15, and up to 30 minutes.
  + Output shape: (3 magnetic field components × number of prediction horizons). Lets try one prediction horizon at 15 min ahead first.

**Implementation Guidelines for Developers**

* Clearly separate preprocessing steps for Branch 1 (time series) and Branch 2 (image data) – this involeves making batches for Branch 1 and Branch 2 by making indexing-lists.
* Maintain modular code structures, particularly for embedding extraction and attention mechanisms, with descriptive headers and comments.
* Utilize Multi-Head Cross-Attention layers for interpretability.
* Document all code thoroughly, especially attention-related modules, to ensure ease of understanding and reproducibility.
* Allow for experimentation by making embedding sizes adjustable.

**Expected Benefits**

* Improved forecasting accuracy through explicit modeling of interactions between global and local geophysical dynamics.
* Robust model performance despite heterogeneous data types.
* Increased interpretability through attention visualization tools, aiding in understanding and diagnosing model predictions.

### 