1 Feature Engineering & Training Data

The game simplifies the 30x20 grid into a 6-feature vector for the neural network. This streamlining allows the neural network to efficiently learn from the most impactful aspects of the game state. Data for training is recorded every tenth game tick, with this data being extracted and saved to two CSV files, game state and expected actions. Normalization standardizes data, enhancing learning efficiency.

Feature	Description
1. Distance Above (Next Column)	Normalized distance above
2. Distance Below (Next Column)	Normalized distance below
3. Obstacle Directly Ahead	Indicator of an immediate obstacle
4. Distance Above (Two Columns Ahead)	Normalized distance above
5. Distance Below (Two Columns Ahead)	Normalized distance below
6. Player Position	Plane's normalized position

2 Network Topology & Configuration

The neural network employs a regression-based approach, as it predicts a continuous value which represents the planes expected movement. The networks architecture is configured with an input layer with 6 nodes, each representing a feature, two hidden layers, with 30 and 15 neurons respectively, both using the ReLU activation function, and an output layer which uses the tanh activation function.

Layer Type	Number of Neurons	Activation Function
Input Layer	6	-
Hidden Layer 1	30	ReLU
Hidden Layer 2	15	ReLU
Output Layer	1	Tanh

The ReLU activation function was chosen for the hidden layers due to its simplicity. It helps the network learn faster without a high computational cost, while also ensuring our network is less prone to overfitting. The tanh function was chosen for the output layer as it outputs a value of between -1 and 1, which is perfect for mapping our planes movement. The amount of neurons were chosen as a balance between efficiency and complexity, to ensure the network can perform well, without being slow.

3 Training Hyperparameters

- **Learning Rate (0.001):** Low value enables the network to make minute adjustments, avoiding overstepping boundaries.

- **Momentum (0.9):** Allows the plane to smoothly change it's trajectory around potential obstacles.
- **Epochs (50,000):** Ensures the model has thorough time to learn, allowing efficient pattern recognition, even in complex situations.
- **Loss Function (SSE):** SSE focuses on correcting large errors, encouraging the network to make precise predictions.

4 Testing/Validation

To test the network, the networks parameters were iteratively tuned to different values, before the network was re-trained. Each network was allocated 3 autopilot runs, with the average time being used to analyze its efficiency. Models from each iteration were compared, highlighting the strengths and weaknesses through targeted testing. Each iteration was trained through a 60 second training period, to ensure a equal and fair amount of data allocation for each model.