

1. Strategy Overview

The Strategy combines **technical indicators** with a **Recurrent Neural Network (RNN)** to predict price movements and make trading decisions. The strategy operates on the **SPY ETF** (which tracks the S&P 500 index) and aims to generate profits by dynamically entering and exiting positions based on both traditional technical signals and predictions from an RNN.

This strategy uses the following key elements:

- **Technical Indicators:** RSI (Relative Strength Index), SMA (Simple Moving Averages).
- **RNN Model:** A Recurrent Neural Network that uses recent historical data to predict future price changes.
- **Feature Extraction:** Price change, volume change, and overnight gap are the key features input to the RNN.

2. Technical Indicators Used

- **RSI (Relative Strength Index):**
 - A **momentum indicator** used to measure the speed and change of price movements.
 - In this strategy, a **2-period RSI** is used to identify **overbought** and **oversold** conditions.
 - RSI(2) below 10 indicates the market is **oversold**, and above 90 suggests the market is **overbought**.
- **SMA (Simple Moving Averages):**
 - **SMA(9):** A short-term moving average to help identify short-term price trends.
 - **SMA(200):** A long-term moving average to help identify the overall trend of the market. Prices above the 200-day moving average indicate an uptrend, while prices below it indicate a downtrend.

The combination of these indicators provides a basic framework for entering long or short positions based on technical signals.

3. Recurrent Neural Network (RNN)

The **RNN** is the heart of the strategy's predictive power. It processes sequences of historical data to predict future price changes. Here's a breakdown of the key components of the RNN and how it works:

RNN Overview:

- **Recurrent Neural Networks (RNNs):** Unlike traditional feedforward neural networks, RNNs have loops in their architecture that allow them to maintain a "memory" of past inputs. This makes RNNs particularly well-suited for tasks involving sequential data, such as time series predictions or stock price forecasting.
- **Why RNN:** Stock prices form a time series, meaning that the order in which data points (prices, volumes) occur is important. RNNs can remember and capture dependencies in this sequence, allowing the model to learn meaningful patterns over time.

4. Training and Weight Optimization

The goal of **training the RNN** is to find the optimal weights that allow the model to make accurate predictions based on input features. The process can be divided into two key phases: **training** and **testing**.

Training Phase:

- During training, the RNN processes historical price data (from 2000 to 2009 in this case) and adjusts its weights (parameters) to minimize the error in its predictions.
- The initial weights in the model are set **randomly**. These weights are:
 - **W_{xh}**: Weights connecting the input layer to the hidden layer.
 - **W_{hh}**: Recurrent weights connecting hidden states across time steps.
 - **W_{hy}**: Weights connecting the hidden state to the output layer.
- The training process involves the following steps:
 1. **Forward Pass**: The RNN processes a sequence of input features (price changes, volume changes, overnight gap) over the lookback period (10 days).
 - The input is transformed into a hidden state at each time step, and the final hidden state is used to make a prediction about future price movements.
 2. **Loss Calculation**: The model calculates the error (difference) between the predicted price and the actual price.
 3. **Backward Pass (Backpropagation Through Time)**: The model calculates gradients of the loss with respect to each weight, which determines how much each weight should be adjusted to reduce the error.
 4. **Weight Updates**: The weights are updated using **gradient descent**, where the learning rate controls how fast the weights are adjusted. The goal is to reduce the prediction error over time.

Testing Phase:

- Once the model has been trained (i.e., the weights are optimized), the model can be used to make predictions on new, unseen data (from 2010 to 2020 in this case).
- During this phase, the model no longer updates its weights. Instead, it uses the optimized weights to predict future prices based on recent market conditions.

Key Hyperparameters:

- **Hidden Size**: The number of units in the hidden layer (64 in this strategy). This controls how much information the model can capture at each time step.
- **Learning Rate**: Controls how quickly or slowly the model adjusts its weights (0.01 in this case). A higher learning rate leads to faster but potentially less stable learning.
- **Lookback Period**: The number of past days used as input to the RNN (10 days in this strategy). It controls how much historical data the model considers when making predictions.

5. Feature Extraction for the RNN

The model uses **three features** from historical data as inputs:

1. **Price Change**: The percentage change in the closing prices between consecutive days.

2. **Volume Change:** The percentage change in trading volume between consecutive days.
3. **Overnight Gap:** The percentage difference between the opening price of the current day and the closing price of the previous day.

These features are computed for the past 10 days (lookback period) and are fed into the RNN to generate a prediction about the future price movement.

6. Trading Logic and Rules

Once the RNN generates a prediction, the model combines the prediction with signals from the technical indicators (RSI and moving averages) to make trading decisions:

- **Long Entry Conditions:**
 - The RSI(2) is below 10 (indicating the market is oversold).
 - The price is above the 200-day moving average (indicating a long-term uptrend).
 - The RNN predicts positive price movement (prediction > 0).
 - If all conditions are met, the strategy enters a long position in SPY.
- **Long Exit Conditions:**
 - The strategy exits the long position when the price rises above the 9-day moving average or the RNN prediction turns negative.
- **Short Entry Conditions:**
 - The RSI(2) is above 90 (indicating the market is overbought).
 - The price is below the 200-day moving average (indicating a long-term downtrend).
 - The RNN predicts negative price movement (prediction < 0).
 - If all conditions are met, the strategy enters a short position in SPY.
- **Short Exit Conditions:**
 - The strategy exits the short position when the price falls below the 9-day moving average or the RNN prediction turns positive.

7. Predictions and Decision-Making

The RNN's prediction plays a key role in enhancing the decision-making process:

- **Positive Prediction:** Indicates an expected price increase, which triggers a long position if other conditions (RSI and MA) are met.
- **Negative Prediction:** Indicates an expected price decrease, which triggers a short position if other conditions are met.

The combination of RNN predictions and technical indicators aims to create a more adaptive strategy that can respond to both market conditions and potential price patterns.