Presentation Outline: Predicting Stock Trends Using Neural Networks

1 Statement of the Problem

Investors aim to anticipate market movements to make more informed trading decisions. We aim to build a model that predicts the short-term trend (up, neutral, or down) of Apple Inc. (AAPL) stock, using historical return data. This model could assist with portfolio timing and risk management.

2 Data Processing

We collected historical stock price data for Apple Inc. (AAPL) from Yahoo Finance, covering the period from 2012 to 2025.

We calculated daily percentage returns using the formula:

$$\text{Return}_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

Then, we created sliding windows of 15 consecutive returns to form input features. For each sample, we summed the next 5 daily returns and assigned a label based on the following rule:

- If the sum $>$ 0.3 $\times$ std $\rightarrow$ Uptrend (label = 2)
- If the sum $<$ -0.3 $\times$ std $\rightarrow$ Downtrend (label = 0)
- Otherwise $\rightarrow$ Neutral (label = 1)

This process generated 3,304 labeled samples, each with 15 input features.

3 Description of the Model

We used a feedforward neural network for classification. The model architecture includes:

- Input layer: 15 features (returns)
- Hidden Layer 1: Dense(12) + BatchNormalization + LeakyReLU
- Hidden Layer 2: Dense(20) + LeakyReLU + Dropout(0.25)
- Hidden Layer 3: Dense(8) + BatchNormalization + LeakyReLU
- Output Layer: Dense(3), activation = softmax

Training Parameters:

- Loss function: Sparse categorical crossentropy
- Optimizer: Adam
- Batch size: 16
- Epochs: 50

The dataset used in training consists of 3,304 samples, each containing 15 past return values as features and a trend label (0, 1, or 2).

Example input format:

$$[\text{Change}_1, \text{Change}_2, \ldots, \text{Change}_{15}, \text{Trend}]$$

We split the dataset chronologically:

- 80% of the oldest data for training
- 20% of the most recent data for testing

This approach simulates real-world forecasting by training on past data and predicting future trends.

4 Mathematical Methods

The return for day t is calculated as:

$$\text{Return}_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

Each sample is a 15-day window of return values. The target is a 3-class label based on the sum of future 5-day returns.

Trend classification uses a dynamic threshold:

$$\text{Threshold} = 0.3 \times \text{std}(\text{FutureReturnSum})$$

Labeling rules:

- Sum $>$ threshold $\rightarrow$ label $=$ 2 (uptrend)
- Sum $<$ -threshold $\rightarrow$ label $=$ 0 (downtrend)
- Otherwise $\rightarrow$ label $=$ 1 (neutral)

Leaky ReLU Activation Function

Leaky ReLU is a variation of the ReLU function that allows a small, non-zero gradient when the input is negative:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases} \quad \text{where } \alpha = 0.01$$

Softmax Activation Function

The Softmax function is used in the output layer to convert logits into probabilities:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Each output $\sigma(z_i)$ represents the predicted probability of class i, and all outputs sum to 1.

5 Interpretation of the Results

The model achieved an accuracy of 92.28% and a loss of 0.2228 on the test dataset.

Prediction results closely follow actual trend movements, as visualized in our price vs. trend plot.

Limitations:

- Only historical returns were used—no news, volume, or macroeconomic features
- Neural networks are black-box models and difficult to interpret

Future improvements:

- Add technical indicators (MACD, RSI)
- Experiment with LSTM or GRU for time-sequence modeling
- Apply to other stocks or asset classes