# Neural Networks and OHLC Terminologies

|  |  |  |  |
| --- | --- | --- | --- |
| Term | Definition | Formula | Usage in OHLC |
| Loss Function | A mathematical function that quantifies how incorrect a model's predictions are compared to actual values. | L = f(y\_actual, y\_predicted) | Used to evaluate prediction accuracy of models forecasting stock prices. |
| Mean Squared Error (MSE) | A common loss function that calculates the average of squared differences between actual and predicted values. | MSE = (1/n) \* Σ (y\_actual - y\_predicted)^2 | Measures stock prediction errors by penalizing large deviations. |
| Cross-Entropy Loss | A loss function used for classification problems, measuring how well predicted probabilities match actual labels. | H(P, Q) = -Σ P(x) log Q(x) | Used in classification models to predict bullish/bearish trends. |
| Gradient Descent | An optimization algorithm used to minimize the loss function by adjusting weights iteratively. | θ = θ - α \* ∇L(θ) | Optimizes weight adjustments for minimizing prediction errors. |
| Backpropagation | A method for updating model parameters using gradient descent by propagating errors backward through the network. | dL/dθ = dL/dy \* dy/dθ | Ensures weight updates improve stock price predictions. |
| Weights | Numerical values assigned to each input feature, determining its influence on predictions. | y\_pred = w1\*x1 + w2\*x2 + ... + wn\*xn + b | Determines the impact of OHLC values on the model's predictions. |
| Bias | A constant added to the model's predictions to account for systematic differences. | b remains constant and is updated alongside weights | Offsets systematic errors in stock price predictions. |
| Activation Function | A function applied to neuron outputs to introduce non-linearity (e.g., ReLU, Sigmoid, Tanh). | ReLU(x) = max(0, x), Sigmoid(x) = 1/(1+e^(-x)) | Introduces non-linearity to capture complex OHLC patterns. |
| Recurrent Neural Network (RNN) | A neural network designed for sequence data, where outputs depend on previous time steps. | h\_t = f(W\*h\_(t-1) + U\*x\_t + b) | Captures sequential patterns in stock price movements. |
| Long Short-Term Memory (LSTM) | A type of RNN that maintains long-term dependencies by using memory cells and gates. | i\_t = σ(W\_i\*x\_t + U\_i\*h\_(t-1) + b\_i), f\_t = σ(W\_f\*x\_t + U\_f\*h\_(t-1) + b\_f) | Handles long-term dependencies in stock price trends. |
| Gated Recurrent Units (GRU) | A variation of LSTM with fewer parameters, making it faster and more efficient. | GRU uses update and reset gates instead of separate forget/input gates | Computes efficient stock trend predictions with lower complexity. |
| Convolutional Neural Network (CNN) | A neural network designed to extract features from grid-like data using convolutional layers. | Y = Conv(X) + Bias | Identifies stock patterns (e.g., candlestick formations). |
| Transformer Model | An advanced deep learning model using self-attention mechanisms for sequence prediction. | Attention(Q, K, V) = softmax(QK^T / sqrt(d\_k)) V | Enhances stock price forecasting by capturing long-range dependencies. |
| Autoencoder | A neural network that learns to encode and reconstruct data, used for anomaly detection. | X\_reconstructed = Decoder(Encoder(X)) | Detects unusual stock movements by reconstructing OHLC data. |
| Reinforcement Learning (RL) | A method where an agent learns to make optimal decisions through trial and error. | Reward maximization using Q-learning or policy gradients | Develops AI-driven trading strategies based on historical stock behavior. |
| OHLC Data | A financial dataset containing Open, High, Low, and Close prices for a given time frame. | OHLC Data: {Date, Open, High, Low, Close} | Provides structured financial data for analysis and modeling. |
| Simple Moving Average (SMA) | A technical indicator that smooths price fluctuations by averaging over a specific period. | SMA = (P\_t + P\_(t-1) + ... + P\_(t-n+1)) / n | Smooths price fluctuations to reveal trends in stock prices. |
| Cosine Similarity | A similarity measure that compares vectors by computing the cosine of the angle between them. | cos(θ) = (A·B) / (||A|| \* ||B||) | Measures how similar two OHLC-based stock movements are. |
| Singular Value Decomposition (SVD) | A dimensionality reduction technique that decomposes a matrix into singular values. | A = UΣV^T | Reduces complexity of OHLC data while retaining essential patterns. |
| Bayesian Inference | A probabilistic approach for updating beliefs based on new evidence. | P(H|D) = (P(D|H) \* P(H)) / P(D) | Adjusts probability estimates for stock trends based on new data. |