Using Deep Learning Neural Network to identify trading opportunities in the e-Mini futures market, and automatically execute trade orders to achieve optimal P/L in day trading

DLNN

Automatic Trading System

Deep Learning neural network for automatic trading

Table of Contents

[Introduction 4](#_Toc40013988)

[Project Goals 4](#_Toc40013989)

[price actions Historical Data Specifications 4](#_Toc40013990)

[Market indicators 6](#_Toc40013991)

[simple moving average (sma) 6](#_Toc40013992)

[sma usage 6](#_Toc40013993)

[sma Calculation 6](#_Toc40013994)

[relative strength index (rsi) 6](#_Toc40013995)

[rsi usage 6](#_Toc40013996)

[rsi calculation 6](#_Toc40013997)

[different trading opportunities and identification 7](#_Toc40013998)

[Reversal 7](#_Toc40013999)

[simple continuation 8](#_Toc40014000)

[resistance and support (R/S) continuation 8](#_Toc40014001)

[what are resistance and support levels 9](#_Toc40014002)

[example 9](#_Toc40014003)

[Special note 10](#_Toc40014004)

[Trade order entry and exit 10](#_Toc40014005)

[Special Note 10](#_Toc40014006)

[price actions chart time frame explanation 10](#_Toc40014007)

[time based chart 10](#_Toc40014008)

[tick based chart 11](#_Toc40014009)

[special note 11](#_Toc40014010)

[Training Data Specifications 11](#_Toc40014011)

[do DLNN Trainings need market indicator data? 11](#_Toc40014012)

[training data format 11](#_Toc40014013)

[trade execution 11](#_Toc40014014)

[integration and deployment details 11](#_Toc40014015)

[ninjatrader trading platform 11](#_Toc40014016)

[ninjascript program 12](#_Toc40014017)

[Division of work between dlnn and ninjascript 12](#_Toc40014018)

[Programming addon for ninjatrader 12](#_Toc40014019)

[Future Improvements 12](#_Toc40014020)

[tuning profit-taking/stop-loss (PT/SL) 12](#_Toc40014021)

[Auto tunning of profit-taking/stop-loss (PT/SL) ratio 12](#_Toc40014022)

[continuous profit taking until market turned 13](#_Toc40014023)

[tuning market indicators 13](#_Toc40014024)

[sma – TBD 13](#_Toc40014025)

[rsi – TBD 13](#_Toc40014026)

[additional market indicators 13](#_Toc40014027)

[market profile indicator – TBD 13](#_Toc40014028)

[depth of market (dom) indicator – TBD 13](#_Toc40014029)

[dadditional trading opportunities identification 13](#_Toc40014030)

[trading different time frames 13](#_Toc40014031)

[trading different markets 13](#_Toc40014032)

# Introduction

In an intelligent automatic trading system, the Deep Learning Neural Network upon identifying a trading opportunity will issue either a long buy or a short sell trade instruction, depending on the detected market condition, to the NinjaScript program running in NinjaTrader platform for trade execution. The NinjaScript program running in the NinjaTrader platform, in turn carries out the actual order executions through the broker. The automatic trading system is initially designed for e-Mini market daytrading, it can be extended to trade in different time horizons and for different markets.

# Project Goals

1. Create a Deep Learning Neural Network (DLNN) to trade in E-Mini futures market.
2. DLNN will automatically identify TRADING OPPORTUNITIES.
3. Once a trading opportunity is identified, automatically enters trade.
4. During the continuous trading hours in a day (no trade should carry over to next day to avoid Gap Up and Gap Down), achieve optimal cumulative profits for all the trades in a day.

Each trade (BUY/SELL order) is entered with "good" Profit-taking/Stop-loss ratio. For example, there are only TWO exit conditions when entering a trade, that is, it will EITHER take-profits at X ticks OR stop-loss at Y ticks. This Profit-taking/Stop-loss ratio (PT/SL) should be adjustable, e.g. it can be adjusted to take-profits at 8 ticks/stop-loss at 4 sticks, depending on user decision based on market volatility.

Market volatility or VIX is a technical indicator that can be calculated from price actions data.

We will initially focus ONLY on two TRADING OPPORTUNITIES, namely Reversals (overbought and oversold) and Continuation on an established trend (price action takes a break before continuing on an upward or downward trend). In the future, we can introduce more trading opportunity scenarios and further improve the DL network.

# price actions Historical Data Specifications

There are only five types of data in the historical price actions data file from the data provider, namely

1. Date and time – the date and time in which a trade was transacted between a buyer and a seller. Note that the time granularity goes down to the milliseconds.
2. Last transacted price – the price at which a trade was last transacted, it could be the same as bid price or ask price, lower than the Bid price (indicating price action trending down), or higher than the Ask price (indicating price action trended up).
3. Bid price – price offered by the buyers, typically lower than the Ask price.
4. Ask price – price offered by the sellers, typically higher than the Bid price.
5. Volume transacted at the last price – the volume in which the last trade transaction took place, higher volume indicates significant movement of the price actions (indicating up or down trend depending on the Last transacted price is Bid or Ask price, or even higher than Bid or lower than Ask price).

The format of the historical data text file is as follows: it begins with Date, follow by Time HHMMSS, FFFFFFFF (time goes down to the milliseconds, hence the last four zeros can be ignored), Last transacted price, Bid price, Ask price, and lastly, the Volume transacted at the last price. The historical data can be downloaded from my data provider, however the Level 2 data is missing.

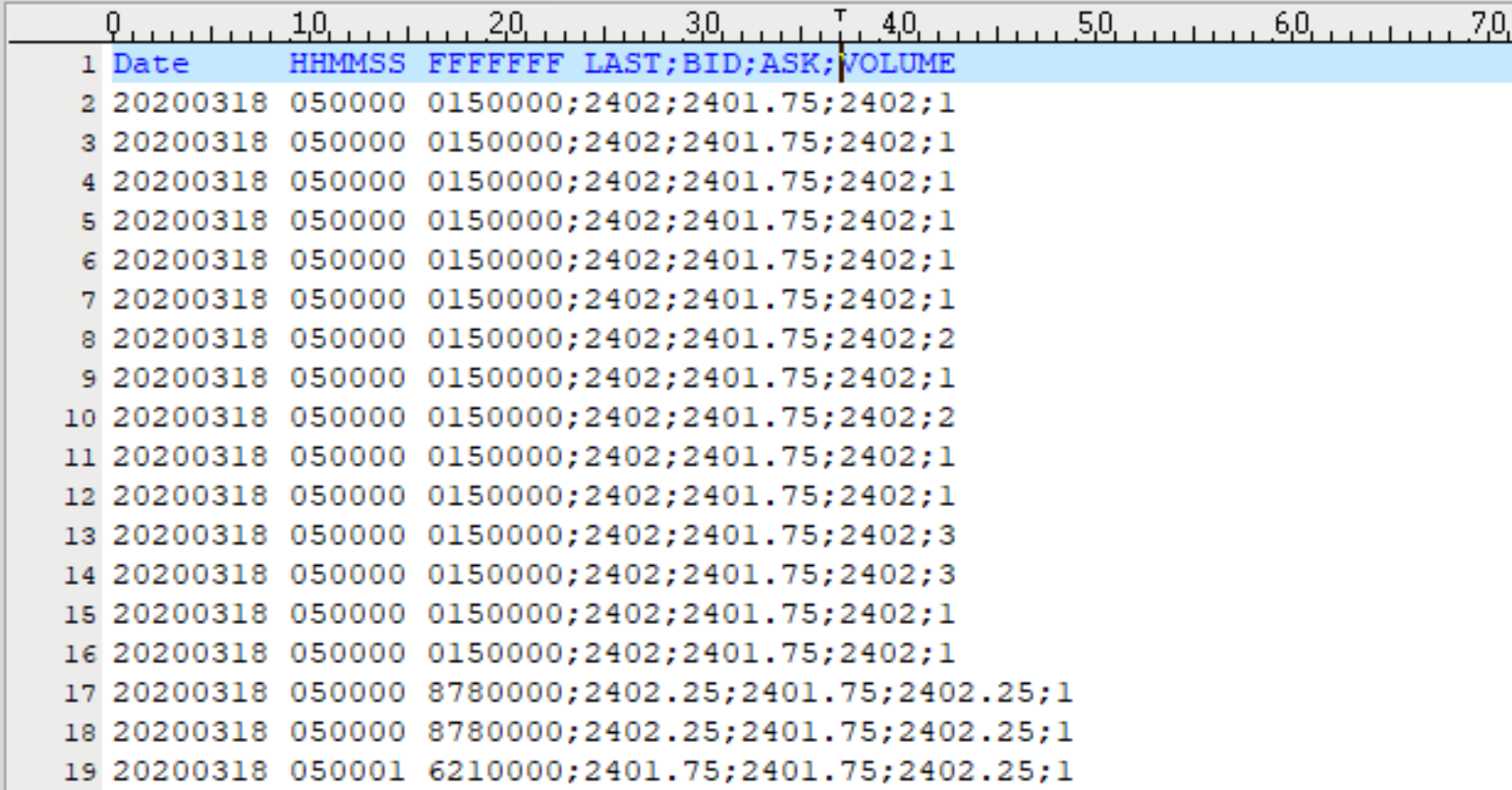


Figure 1 example of price actions transaction data format

Level 2 data are the data in the SuperDOM (see picture below), which reflects the market depth (the "relative strength" of the bears and bulls at the time) of the time. The traders are "posturing" at different price levels ready to transact if and when the price hit the price point the traders put their respective buy/sell entries at. For the initial phase of our project, we can probably ignore the Level 2 data for Trading Opportunities identification.

The Level 2 data, if desirable can be collected over time using a programming script.



Figure 2 example of Depth of Market (DOM) table

Below please find a sample of a SINGLE DAY market data readily available for download - note that the file is quite voluminous at 79M for a SINGLE day of data. A year’s worth of historical data will be somewhere around 30G of data, but the historical data set alone may not be sufficient for DL training (will address in subsequent section for further discussion).

[**[https://ci4.googleusercontent.com/proxy/qYa6_H461C21RR-qYIwSBcI7qqS9WSiFAN0DaF9sKsbdOZ5S3C-vrr7s_W7pzqFo7qGPHn6zXYPCgzf3sLpi6Csr9HA9gNV6rWRnwUrSjrNGKy8ubI-d=s0-d-e1-ft#https://ssl.gstatic.com/docs/doclist/images/icon_10_generic_list.png](https://drive.google.com/file/d/13ZloGBNuxgELcwk34Dhznkfkgj8m5QzP/view?usp=drive_web) ES 06-20.Last.txt**](https://drive.google.com/file/d/13ZloGBNuxgELcwk34Dhznkfkgj8m5QzP/view?usp=drive_web)

# Market indicators

## simple moving average (sma)

### sma usage

1. SMA is to indicate trending of the market.
2. When SMA20 and SMA50 are in parallel, the market is consistently trending either upward or downward.
3. When SMA20 crosses SMA50 from below and diverging, the market is temporary trending upwards.
4. When SMA20 crosses SMA50 from above and diverging, the market is temporary trending downwards.

### sma Calculation

The Simple Moving Average (SMA) is calculated by adding the price of an instrument (in our case, the e-Mini market last transacted price) over a number of time periods and then dividing the sum by the number of time periods.

SMA = ∑1..n Price n / n

Where: n = Time Period

Currently we use two SMA indicators, SMA20 and SMA50, where n=20 and 50 respectively.

## relative strength index (rsi)

### rsi usage

1. RSI calculation aka relative strength index is an oscillating indicator that is displayed as a line.
2. It moves between two extremes; overbought and oversold.
3. The bullish and bearish price movements are plotted against the instrument price.
4. Compares bullish and bearish price movements and maps them out as an oscillator.
5. The calculated range of the RSI is 0-100.
6. Oversold levels are between 0 and 30, during oversold the market is poised for reversal upwards.
7. Overbought areas are between 70-100, during overbought the market is poised for reversal downwards.

### rsi calculation

To simplify the calculation, RSI is broken down into its basic components: RS, which is calculated from Average Gain and Average Loss. This RSI calculation is based on 14 periods, which is the default. Losses are expressed as positive values, not negative values.

100

RSI = 100 - --------

1 + RS

Where RS = Average Gain / Average Loss

Initially, the calculations for average gain and average loss are simple 14-period averages:

***First Average Gain = Sum of Price Gains over the past 14 periods / 14***

***First Average Loss = Sum of Price Losses over the past 14 periods / 14***

Subsequently, calculations are based on the prior averages and the current Price gain and loss:

***Average Gain = [(previous Average Gain) x 13 + current Gain] / 14.***

***Average Loss = [(previous Average Loss) x 13 + current Loss] / 14.***

# different trading opportunities and identification

## Reversal

1. Reversal happens when RSI crosses 30, that is the market has an oversold signal OR,
2. Conversely, RSI crosses 70, that is the market has an overbought signal, AND
3. Provided the market is NOT consistently trending upwards or downwards, that is SMA20 and SMA50, are not parallel to each other, AND
4. In the case of oversold signal, DLNN enters a Long trade (buy) when the current RSI number is higher than the last two RSI numbers, i.e. the price actions is now reversing upwards, AND
5. Conversely, in the case of overbought signal, DLNN enters a Short trade (sell) when the current RSI number is lower than the last two RSI numbers, i.e. the price actions is now reversing downwards, AND
6. Provided this is not a reversal trade with identical signal from a previously failed reversal trade. For example, if DLNN traded reversal based on an oversold signal and resulted in a loss, the system should NOT attempt ANOTHER reversal trade base on the repeated oversold signal. Instead, the system should wait for an overbought signal to carry out the next reversal trade.



Figure 3 an example of a reversal trade base on an oversold signal.



Figure 4 an example where a reversal trade based on overbought signal resulted in a loss, the next reversal trade should NOT be based on repeated identical overbought signal.

## simple continuation

1. Continuation happens when the RSI crosses the middle 50 line, AND
2. The DLNN enters a Long trade (buy) when the RSI crosses from lower value to higher value after crossing the 50 middle line and the current RSI number is higher than the last two RSI numbers, AND
3. Conversely, the DLNN enters a Short trade (sell) when the RSI crosses from higher value to lower value after crossing the 50 middle line and the current RSI number is lower than the last two RSI numbers.



Figure 5 continuation trade when RSI crosses the 50 middle line.

## resistance and support (R/S) continuation

### what are resistance and support levels

In stock market technical analysis, support and resistance are levels of the price actions in which trading prices will tend to stop and reverse. These levels are denoted by multiple touches of price without a breakthrough of the level. For our purpose, we define these levels as the price levels with the recorded HIGHER transaction volumes for the day.

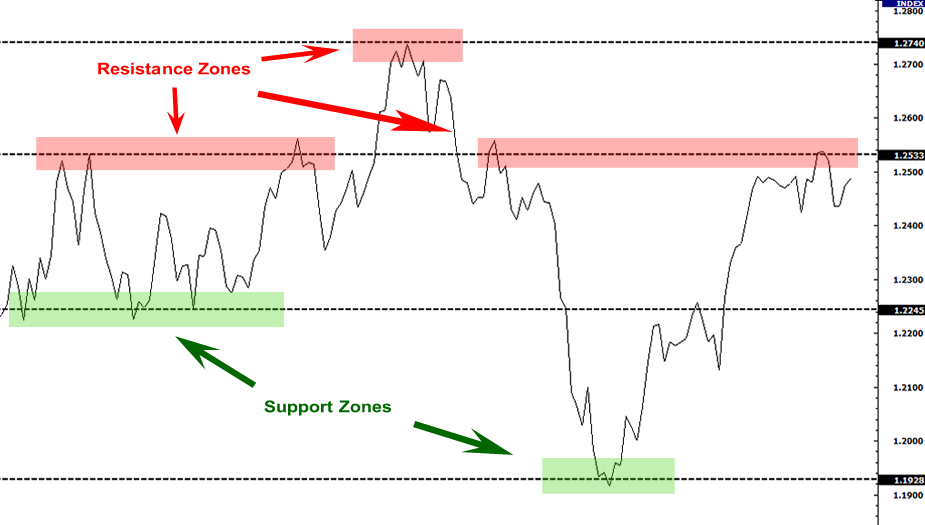


Figure 6 example of resistance and support

### example

1. Resistance and Support Levels Continuation happens when a market is starting to trend either upwards or downwards, that is SMA20 and SMA50 are starting to diverge, AND
2. The price actions initially “stuck” and traded sideways either at a resistance or support level. Once the price broke through the support level, price actions will head lower to the next trading range. Conversely, if the price broke through the resistance level, price actions will head higher to the next trading range, AND
3. DLNN enters a Long trade (buy) at the point of breakthrough of the resistance level, OR
4. DLNN enters a Short trade (sell) at the point of breakthrough of the support level.



Figure 7 SMA 20 and 50 briefly converged and started diverging, the price actions after experiencing support level and finally broke through to a lower price level



Figure 8 Price actions breakthrough at the resistance level

### Special note

Resistance and Support Continuation trade is a more sophisticated trade pattern, we may want to consider leaving it for future improvements if this proves to be too difficult to learn by the DLNN.

# Trade order entry and exit

1. When a trade opportunity is identified, a predefined order quantity of trade order will be placed.
2. A trade order will also specify exit conditions (common approach), namely Profit Taking and Stop Loss prices.
3. Profit Taking specifies the price point in which the broker will automatically exit the trade by selling (if entered via Long buying) or buying (if entered via Short selling) at the market price, at an equivalent order quantity, and taking profits as a result.
4. Conversely, Stop Loss specifies the price point in which the broker will automatically exit the trade by selling (if entered via Long buying) or buying (if entered via Short selling) at the market price, at an equivalent order quantity, and taking a loss as a result. Stop Loss is to prevent further damage to a trade position going sour. We should NEVER enter a trade without Stop Loss.

## Special Note

Alternatively, we can omit specifying profit taking and let DLNN decides when to take profits, this will potentially extend a winning trade. We will address this possibility in the Future Improvements section.

# price actions chart time frame explanation

## time based chart

A user can use different ways to plot a price actions chart. For example, some use 1 minute, 15 minutes or even 1 hour between each candlestick bar to plot the price actions chart. The shorter the period, the more sensitive the chart will be to reflect market dynamics, hence 1-minute chart is popular amongst day traders. However, sometimes even the 1-minute chart is not sufficient to reflect the market dynamics because very high volume transactions can happen within a minute.

## tick based chart

To achieve better granularity and reflect market dynamics, we use tick counts to plot the price actions chart instead. A tick count price actions chart is plotted such that each candlestick bar is drawn only when a predefined numbers of transactions (tick counts) had taken place. For our purpose, we plot 2000 tick counts for each candlestick bar, therefore every candlestick bar represents 2000 trade transactions.

## special note

Perhaps the DLNN does not care about how we plot our chart, because the price actions data format remains the same regardless, as described in the Price Actions Historical Data Specification:

Date, Time HHMMSS, Last transacted price, Bid price, Ask price, and Volume

# Training Data Specifications

## do DLNN Trainings need market indicator data?

Market indicators are essentially for human interpretation of the market conditions, does a DLNN training data set needs to include the calculations of corresponding SMA and RSI values? Is the price actions data set lone sufficiently captured all the market behaviors we seek to identify? What additional data are needed in order for a DLNN to learn in order to identify the trade opportunities?

## training data format

1. We will provide daily price actions data file as individual training data set. How many days of training data set is needed?
2. There will be multiple trading opportunities in a single day, how do we annotate these trading opportunities in the training data set such that our DLNN can learn to identify these trading opportunities?

## trade execution

As soon as a trade opportunity is identified, DLNN can issue one of the following trade instructions to the trade execution module (NinjaScript) for trade executions (see next section for integration with broker for execution):

* Reversal Long (RL) – buy long when Reversal opportunity is identified.
* Reversal Short (RS) – sell short when Reversal opportunity is identified.
* Simple Continuation Long (SCL) – buy long when Simple Continuation opportunity is identified.
* Simple Continuation Short (SCS) – sell short when Simple Continuation opportunity is identified.
* R/S Continuation Long (RSCL) – buy long when Resistance/Support Continuation opportunity is identified.
* R/S Continuation Short (RSCS) – sell short when Resistance/Support Continuation opportunity is identified.
* Profit Taking (PT) – refer to Future Improvements section for details.

# integration and deployment details

## ninjatrader trading platform

We will be using NinjaTrader Trading Platform to integrate with broker and execute trade orders. NinjaTrader comes with Free and Paid versions, and it can be downloaded at <https://ninjatrader.com/LP/NT8_Future/Introducing-NinjaTrader-8>. We will be using the paid version, with all features available for our project. Initially we will create simulated trading account(s) on NinjaTrader, trading on live e-Mini market data.

## ninjascript program

All automated trade order executions will be issued through a program running in NinjaTrader platform, known as NinjaScript. NinjaScript is a C# language program that runs in the NinjaTrader trading platform. In order for NinjaScript to interact with DLNN module, we will need to call out to a DLNN Wrapper function from the NinjaScript program. The details will have to be worked out in the future, however because we are NOT a high frequency trading system, performance should not be an issue.

## Division of work between dlnn and ninjascript

While it is clear that DLNN will identify trade opportunities and issue trade instructions to NinjaScript for order executions, it is unclear if NinjaScript should maintain some forms of intelligence of its own. The following are some considerations:

* NinjaScript should automatically stop all trading activities for the day if the incurred losses surpassed the daily limit. This feature is probably not important during the testing phase, but this feature is essential for risk management in an actual deployment.
* Perhaps we could implement the continuous profit taking (see Future Improvements section for further details) in the NinjaScript because it is where daily P/L can be easily tracked?
* Should DLNN focus on identification of trade opportunities and leave the refined order placement heuristics to NinjaScript? For example varying PT/SL ratio based on real time VIX indicator, or disallowing consecutives identical Reversal trade instructions if the previous similar Reversal trade resulted in a loss.
* Additional design issues will need to be discussed further in order to maximize the strength of DLNN and leave the more mundane tasks to NinjaScript.

## Programming addon for ninjatrader

<https://ninjatrader.com/support/helpGuides/nt8/?developing_indicators.htm>

<https://ninjatrader.com/support/helpGuides/nt8/?addon_development_overview.htm>

# Future Improvements

## tuning profit-taking/stop-loss (PT/SL)

### Auto tunning of profit-taking/stop-loss (PT/SL) ratio

We can experiment with different PT/SL ratios when we trade with a simulated account, in fact we can run multiple simulated accounts each with its own PT/SL ratio. A typical PT/SL ration for day trading might be 10/5, 8/4 or 6/3, depending on the market volatility. We always want to have higher Profit Taking value than Stop Loss, so even if we achieve 50% success rate in trading, our P/L will still be positive.

### continuous profit taking until market turned

It is a common practice to keep a winning trade running until the trend runs out. In this scenario, when we enter a trade position, instead of specify profit taking price point, we let the system decide when to exit the trade. This way, when a trend persist over a long period, one winning trade alone can accumulate substantial profits. This could be relatively easy to implement, and we should consider incorporating this feature in our initial roll out of the DLNN system.

## tuning market indicators

### sma – TBD

### rsi – TBD

## additional market indicators

### market profile indicator – TBD

### depth of market (dom) indicator – TBD

## dadditional trading opportunities identification

[**[](https://drive.google.com/file/d/1b5-CZdMYGN7iMzlZgZNm3jlkyK1l1pgW/view?usp=drive_web) Most Effective Day Trading Strategies and Order...**](https://drive.google.com/file/d/1b5-CZdMYGN7iMzlZgZNm3jlkyK1l1pgW/view?usp=drive_web)

## trading different time frames

TBD

## trading different markets

TBD