Dự đoán giá của cổ phiếu bằng cách nào?

Đầu tiên em tìm hiểu được về các khái niệm liên quan đến giải thuật trên:

Decision tree (Cây quyết định) là gì?

Cây quyết định là một mô hình máy học phi tuyến tính được sử dụng để giải quyết các vấn đề phân loại và dự đoán trong các lĩnh vực khác nhau như kinh doanh, y tế, khoa học dữ liệu và nhiều lĩnh vực khác. Cây quyết định được xây dựng dựa trên các quyết định logic và các quy tắc được trích xuất từ dữ liệu đào tạo.

Cây quyết định bao gồm một số nút, các cạnh và các nút lá. Nút gốc đại diện cho toàn bộ tập dữ liệu và các nút lá đại diện cho các phân lớp hoặc các kết quả dự đoán khác nhau. Các nút trung gian đại diện cho các quyết định phân tách dữ liệu thành các nhóm nhỏ hơn dựa trên các đặc tính của dữ liệu.

Random Forests (Rừng ngẫu nhiên) là gì?

Rừng ngẫu nhiên – Random Forests, *đây là một dạng nâng cao của Cây quyết định* – decision tree; Rừng ngẫu nhiên là một thuật toán học máy linh hoạt, dễ sử dụng , tạo ra kết quả tuyệt vời ngay cả khi không điều chỉnh siêu tham số. “Rừng ngẫu nhiên là một bộ phân loại chứa một số cây quyết định trên các tập con khác nhau của tập dữ liệu đã cho và lấy giá trị trung bình để cải thiện độ chính xác dự đoán của tập dữ liệu đó.”

GA-Genetic Algorithm (Thuật toán di truyền) là gì?

Giải thuật di truyền (GA-Genetic Algorithm) là kỹ thuật phỏng theo quá trình thích nghi tiến hóa của các quần thể sinh học dựa trên học thuyết Darwin. GA là phương pháp tìm kiếm tối ưu ngẫu nhiên bằng cách mô phỏng theo sự tiến hóa của con người hay của sinh vật. Tư tưởng của thuật toán di truyền là mô phỏng các hiện tượng tự nhiên, là kế thừa và đấu tranh sinh tồn.

Nói ngắn gọn GA là phương pháp tìm kiếm tối ưu ngẫu nhiên bằng cách mô phỏng theo sự tiến hóa của con người hay của sinh vật.

Recurrent neural network - RNN là gì?

RNN (Recurrent Neural Network) là một kiến trúc mạng nơ-ron nhân tạo đặc biệt được sử dụng cho các tác vụ liên quan đến chuỗi dữ liệu hoặc dữ liệu thời gian. Trong RNN, các đầu vào không chỉ được xử lý một cách độc lập mà còn được xử lý theo trình tự, trong đó các đầu ra trước đó được sử dụng để tính toán đầu ra hiện tại. Điều này cho phép RNN có khả năng hiểu được các mẫu dữ liệu phức tạp trong chuỗi thời gian, bởi vì thông tin từ các đầu vào trước đó được lưu trữ và sử dụng khi xử lý các đầu vào tiếp theo.

Artificial Neural Network – ANN (Mạng neural nhân tạo) là gì?

Mạng neural nhân tạo (Artificial Neural Network) là một mô hình tính toán được lấy cảm hứng từ cấu trúc và hoạt động của hệ thống thần kinh sinh học trong não người. Mô hình này được xây dựng dựa trên các phương trình toán học và sử dụng một số lượng lớn các "neuron" nhân tạo để xử lý thông tin đầu vào và cho ra kết quả đầu ra.

Các mạng neural nhân tạo có thể được sử dụng để giải quyết nhiều vấn đề khác nhau, bao gồm nhận dạng hình ảnh, dịch văn bản, nhận dạng giọng nói, dự đoán thời tiết, và nhiều ứng dụng khác. Chúng cũng được sử dụng rộng rãi trong các lĩnh vực như khoa học dữ liệu, trí tuệ nhân tạo, robot học, và nhiều lĩnh vực khác.

Long short-term memory là gì?

LSTM là viết tắt của "Long Short-Term Memory", là một kiểu *mô hình mạng nơ-ron nhân tạo (artificial neural network)* được sử dụng phổ biến trong việc xử lý các chuỗi dữ liệu, nhất là trong lĩnh vực xử lý ngôn ngữ tự nhiên (natural language processing) và nhận dạng giọng nói (speech recognition).

Khác với các mô hình truyền thống, LSTM được thiết kế để có khả năng "ghi nhớ" thông tin trong một thời gian dài và "quên" những thông tin không quan trọng. Điều này giúp LSTM có thể xử lý được các chuỗi dữ liệu có độ dài lớn và có các mối quan hệ phức tạp giữa các thành phần trong chuỗi.

LSTM được cấu thành bởi các "cổng" (gate) để điều khiển luồng thông tin, bao gồm cổng "quên" (forget gate), cổng "đầu vào" (input gate) và cổng "đầu ra" (output gate). Các cổng này giúp LSTM lựa chọn thông tin quan trọng để "ghi nhớ" và giữ lại

trong bộ nhớ của mô hình, và lựa chọn thông tin để "quên" hoặc "đưa ra" để đưa ra kết quả.

Adaptive function (Hàm thích nghi) là gì?

Hàm thích nghi (adaptive function) là một khái niệm trong lý thuyết tối ưu hóa, nó mô tả khả năng của một hệ thống để thích nghi với môi trường để đạt được mục tiêu tối ưu. Đối với một hàm thích nghi, hệ thống sẽ thích nghi với môi trường bằng cách thay đổi các tham số hoặc quyết định của nó để đạt được hiệu quả tối ưu hơn.

Intersection method (Phương pháp giao cắt nhiều điểm) là gì?

Phương pháp giao cắt nhiều điểm (Intersection method) là một trong các phương pháp để giải các hệ phương trình phi tuyến tính. Đây là một phương pháp lặp, có thể được sử dụng để giải các hệ phương trình phi tuyến tính đa biến.

Phương pháp này bắt đầu với một tập hợp các đầu vào ban đầu và các giá trị tương ứng của các biến đầu ra. Sau đó, phương pháp sử dụng các phương trình để tính toán giá trị mới cho các biến đầu ra dựa trên giá trị hiện tại của các biến đầu vào. Quá trình này được lặp lại cho đến khi các giá trị của biến đầu ra không thay đổi đáng kể nữa.

Bit mutation (Phương pháp đột biến bit) là gì?

Phương pháp đột biến bit (bit mutation) là một trong những phương pháp cơ bản trong các thuật toán tối ưu hóa tiến hóa (evolutionary optimization algorithms), đặc biệt là trong giải thuật di truyền (genetic algorithm).

Trong phương pháp này, các cá thể (individuals) được biểu diễn dưới dạng chuỗi bit (bitstring) và các bit được đảo ngược ngẫu nhiên với một xác suất nhất định để tạo ra một cá thể mới. uy nhiên, xác suất đột biến thường rất nhỏ, thường chỉ trong khoảng từ 0.01 đến 0.1, để đảm bảo rằng quá trình tối ưu hóa tiến hóa diễn ra một cách ổn định và không mất đi các giá trị tốt của các cá thể đã tìm được.

Phương pháp đột biến bit giúp đánh giá mức độ đa dạng của các cá thể trong quần thể và có thể tạo ra các giá trị mới tiềm năng, giúp giải thuật tìm được giải pháp tối ưu hơn.

Mean Squared Error -MSE (sai số bình phương trung bình) là gì?

MSE là một phép đo độ chính xác của mô hình trong học máy và thống kê.

MSE tính bằng cách lấy trung bình bình phương của các sai số giữa giá trị dự đoán của mô hình và giá trị thực tế, giá trị MSE càng nhỏ thì mô hình càng chính xác.

Phương pháp của paper

Đầu tiên: Dùng thuật toán GA để giải mã tối ưu hóa các thế hệ trước được lại tạo lại với nhau cuối cùng đưa ra quần thể có thể sử dùng được làm lời giải gần đúng nhất cho bài toán

Quá trình xử lý GA có thể được chia thành bảy giai đoạn:

- Khởi tạo quần thể: Tạo ra một tập hợp các cá thể (hay còn gọi là gen) ban đầu, được gọi là quần thể. Mỗi cá thể trong quần thể là một giải pháp có thể của bài toán. Các cá thể ban đầu có thể được tạo ngẫu nhiên hoặc dựa trên kiến thức về bài toán.
- Đánh giá độ thích nghi: Đánh giá độ thích nghi của mỗi cá thể trong quần thể. Độ thích nghi có thể được đo bằng hàm mục tiêu, tức là hàm đánh giá hiệu suất của giải pháp.
- Lựa chọn: Chọn ra một số cá thể trong quần thể để tiếp tục trong quá trình tối ưu hóa tiếp theo. Các cá thể được chọn dựa trên độ thích nghi của chúng. Các cá thể tốt nhất sẽ được chọn để tiếp tục trong quá trình tối ưu hóa.
- Lai ghép: Lai ghép các cá thể được chọn. Lai ghép giữa hai cá thể tạo ra các cá thể mới có tính chất kết hợp từ hai cá thể cha mẹ.
- Đột biến: Đột biến một số cá thể trong quần thể. Đột biến là quá trình thay đổi một số đặc điểm của cá thể một cách ngẫu nhiên, giúp đưa ra những giải pháp mới.
- Thay thế: Thay thế các cá thể cũ bằng các cá thể mới được tạo ra trong quá trình lai ghép và đột biến. Quá trình này giúp nâng cao chất lượng của quần thể.
- Kiểm tra điều kiện dừng: Kiểm tra điều kiện dừng để xác định liệu kết quả tối ưu đã đạt được hay chưa. Nếu chưa, quay lại bước 2. Nếu đã đạt được, dừng quá trình tối ưu hóa.

Toàn bộ quá trình của GA.{α1,α2, … ,αN} đại diện cho bộ tính năng ban đầu. Đầu tiên, nó thiết kế mã hóa nhị phân cho mỗi nhiễm sắc thể β đại diện cho một giải pháp tiềm năng cho vấn đề, nghĩa là mã hóa nhị phân của mỗi nhiễm sắc thể đại diện cho mỗi tổ hợp tính năng. Trong giai đoạn khởi tạo, kích thước dân số được đặt cho dân số và dân số ban đầu là ngẫu nhiên {β1,β2, … ,βN} được tạo ra. Sau đó, mức độ phù hợp của từng nhiễm sắc thể được tính toán theo chức năng phù hợp được thiết lập sẵn. Hàm thích nghi là một chỉ số đánh giá được sử dụng để đánh giá hiệu suất của nhiễm sắc thể. Trong GA, định nghĩa về chức năng phù hợp là yếu tố chính ảnh hưởng đến *hiệu suất.*

Các nhiễm sắc thể có hiệu suất cao thì có nhiều khả năng được chọn nhiều lần hơn, trong khi những nhiễm sắc thể có hiệu suất thấp có nhiều khả năng bị loại bỏ hơn

Quá trình trao đổi chéo nhiễm sắc thể và đột biến có ý nghĩa rất lớn đối với GA. Việc trao đổi đoạn tương ứng trong chuỗi nhiễm sắc thể và thay đổi tổ hợp gen để sinh ra đời con mới có lợi là làm tăng tính đa dạng di truyền của quần thể.

LSTM là một RNN sâu đặc biệt, LSTM giúp tăng cường đáng kể dung lượng bộ nhớ của mô hình nhờ cấu trúc đơn vị thần kinh cơ chế cổng đặc biệt của nó và giải quyết vấn đề biến mất độ dốc do chuỗi đầu vào quá dài trong quá trình học của mạng thần kinh tuần hoàn truyền thống.

Mạng LSTM lưu tất cả thông tin trước mỗi bước thời gian trong đơn vị thần kinh của bước thời gian hiện tại và mỗi đơn vị thần kinh được điều khiển bởi cổng đầu vào, cổng quên và cổng đầu ra.

- Cổng đầu vào được sử dụng để kiểm soát thông tin đầu vào của đơn vị thần kinh tại thời điểm hiện tại
- cổng quên được sử dụng để kiểm soát thông tin lịch sử được lưu trữ trong đơn vị thần kinh tại thời điểm trước đó
- Cổng đầu ra được sử dụng để kiểm soát thông tin đầu ra của đơn vị thần kinh tại thời điểm hiện tại. Mục đích của thiết kế này là cho phép mô hình LSTM ghi nhớ có chọn lọc các thông tin lịch sử quan trọng hơn.

Mô hình của paper

Giai đoạn đầu tiên là sử dụng GA để sắp xếp mức độ quan trọng của các yếu tố. Các bước cụ thể như sau:

1 Mã hóa nhị phân của nhiễm sắc thể, khởi tạo ngẫu nhiên của quần thể. Chúng tôi biểu thị dân số GA bằng cách sử dụng pop như sau:

$$POP = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,k} \\ a_{2,1} & a_{2,2} & \dots & a_{2,k} \\ \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & \dots & a_{m,k} \end{bmatrix}$$

Trong ma trận trên, mỗi hàng biểu thị một nhiễm sắc thể (hoặc một bộ lựa chọn tính năng), độ dài nhiễm sắc thể k biểu thị tổng số tính năng), số m biểu thị kích thước dân số, giá trị của ai,j là 0 hoặc 1, 1 đại diện cho lựa chọn, 0 đại diện cho không lựa chọn, trong đó các số dương khác không.

2 Sử dụng phương pháp roulette (quay số) cho hoạt động lựa chọn. Tính toán mức độ phù hợp của từng nhiễm sắc thể trong quần thể. Xác suất của mỗi cá thể được chọn tỷ lệ thuận với độ phù hợp của nhiễm sắc thể và tổng xác suất nhiễm sắc thể được chọn là 1. Trong thuật toán, quần thể được cập nhật một lần mỗi chu kỳ theo xác suất.

3 Phương pháp giao cắt nhiều điểm được sử dụng để thực hiện thao tác trao đổi chéo và các nhiễm sắc thể giữa hai cá thể sẽ được trao đổi với xác suất trao đổi chéo được đặt là 0,8. Trong quy trình thuật toán, một thao tác chéo được thực hiện trên mỗi nhiễm sắc thể trong mỗi chu kỳ. Mục đích là để tạo ra một xác suất ngẫu nhiên. Nếu xác suất ngẫu nhiên nhỏ hơn xác suất chéo thì sẽ thực hiện trao đổi, ngược lại sẽ không có trao đổi.

4 Phương pháp đột biến bit cơ bản được sử dụng để thực hiện thao tác đột biến. Ở những cá thể hiện đại, một gen bị thay đổi với một xác suất nhỏ. Xác suất biến thể được đặt thành 0. 003. Thuật toán tạo ra một xác suất tại một thời điểm. Nếu xác suất ngẫu nhiên nhỏ hơn xác suất giao nhau, biến thể sẽ được thực hiện; nếu không, sẽ không có biến thể nào được thực hiện.

Lặp các bước (**2**) đến (**4**) cho đến khi lặp được 100 lần. Khi kết thúc thuật toán, đã tạo ra một quần thể tối ưu gần với giải pháp tối ưu. Trong bài báo này, tổng số lần xuất hiện của từng yếu tố trong tổng thể được thống kê xếp hạng theo mức độ quan trọng của yếu tố. *Yếu tố xuất hiện càng nhiều lần thì càng quan trọng.*

Giai đoạn thứ hai của nghiên cứu này là giai đoạn tối ưu hóa lựa chọn đặc trưng của mô hình dự đoán cổ phiếu LSTM. Dựa trên xếp hạng tầm quan trọng của yếu tố thu được trong giai đoạn trước, 40, 30, 20, 10 và 5 yếu tố hàng đầu được lấy làm tính năng đầu vào của mô hình LSTM. Bằng cách so sánh các kết quả dự đoán, tổ hợp nhân tố tối ưu được xác định và mô hình tối ưu được so sánh với các mô hình đường cơ sở để xác minh tính ưu việt của mô hình tối ưu hóa được đề xuất trong việc cải thiện độ chính xác của mô hình.

Mô hình sử dùng hàm sai số bình phương trung bình (MSE) làm chỉ số đánh giá mô hình và công thức như sau:

$$\text{MSE}=\frac{1}{\text{m}}\sum_{i=1}^{i}(\text{yi}-\text{y\^{}i})\text{\^{}}2$$

Trong đó,

m là số lượng mẫu.

yi là giá cổ phiếu và.

y^i là giá cổ phiếu dự báo của mô hình.

Kết quả thực nghiệm của paper

Dùng phương pháp trên để thực nghiệm với 2490 dữ liệu lịch sử của ngân hàng xây dựng Trung Quốc và cổ phiếu CSI 300, từ ngày 1 tháng 1 năm 2010 đến ngày 1 tháng 4 năm 2020 đã được thay thế vào mô hình LSTM và dữ liệu được xử lý bằng cách điền và chuẩn hóa trung bình.

Thử nghiệm được đào tạo theo 80% dữ liệu đầu tiên và được thử nghiệm theo 20% dữ liệu cuối cùng, MSE (sai số bình phương trung bình) đã tập hợp các yếu tố ban đầu và tập hợp con 30, 20, 10, 5 yếu tố hàng đầu trong xếp hạng tầm quan trọng của yếu tố đã được sử dụng làm tính năng đầu vào của mô hình LSTM cho các thử nghiệm so sánh.

Trong đó,

Mô hình LSTM rất phù hợp để dự báo thị trường chứng khoán theo chuỗi thời gian do tính dễ nhớ trong thời gian dài.

Mô hình PCA kết hợp với SVM (PCA-SVM) được coi là mô hình dự báo trữ lượng phi tuyến cổ điển có khả năng cải thiện hiệu quả khả năng tổng quát hóa.

Mô hình dự đoán trữ lượng rừng ngẫu nhiên [8] có thể cải thiện hiệu quả độ chính xác của dự đoán sau khi tối ưu hóa tham số.

Mạng thần kinh tái phát dựa trên sự chú ý giai đoạn kép (DA-RNN) là một mô hình dự đoán tuần tự và khai thác tính năng nâng cao.

Mô hình GA-LSTM được đề xuất có thể hoạt động tốt hơn tất cả các mô hình cơ sở trên cả hai bộ dữ liệu.

Trên tập dữ liệu chứng khoán CSI 300, khi 20 tập hợp con yếu tố hàng đầu trong xếp hạng tầm quan trọng của yếu tố được lấy làm tính năng đầu vào của mô hình GA-LSTM được đề xuất, mức độ phù hợp dự đoán đạt cao nhất. MSE đạt tối thiểu 0,0039.

Khuyết điểm của mô hình

Trong mô hình trên đã cho thấy sự kết hợp hoàn hảo để dự đoán cổ phiếu gần đúng nhất. Trong phương pháp này thuật toán GA được đề xuất cho tính năng lựa chọn đặc trưng để chọn ra các yếu tố phù hợp , và kết hợp mô hình học sâu LSTM mô quan hệ giữa các yếu tố và cổ phiếu được dự đoán gần chính xác nhất.

Tuy nhiên mô hình vẫn còn nhiều hạn chế sau:

Dữ liệu đầu vào: Để đưa ra dự đoán chính xác, mô hình cần phải được cung cấp với dữ liệu đầu vào chính xác và đầy đủ. Nếu dữ liệu đầu vào không đủ, hoặc không chính xác, kết quả dự đoán sẽ không chính xác.

Độ tin cậy của mô hình: Độ tin cậy của mô hình dự đoán có thể bị ảnh hưởng bởi nhiều yếu tố, bao gồm số lượng dữ liệu đầu vào, độ phức tạp của mô hình, cách thức xử lý dữ liệu và các tham số của mô hình. Do đó, mô hình không thể đảm bảo tính đúng đắn và chính xác tuyệt đối trong việc dự đoán giá cổ phiếu.

Điều chỉnh các tham số: Mô hình kết hợp GA và LSTM cần được điều chỉnh các tham số phù hợp để đưa ra kết quả dự đoán tốt nhất. Tuy nhiên, việc điều chỉnh các tham số này là khá phức tạp và tốn thời gian

Khả năng áp dụng của mô hình: Mô hình kết hợp GA và LSTM thường chỉ phù hợp với các công ty hoặc thị trường cổ phiếu có tính biến động cao. Vì vậy, không phải lúc nào cũng áp dụng được cho các công ty hay thị trường cổ phiếu khác.

Khả năng dự đoán tương lai: Mô hình kết hợp GA và LSTM thường chỉ đưa ra dự đoán trong tương lai gần và không đảm bảo tính chính xác về dài hạn. Nếu mô hình đưa ra dự đoán không chính xác, nó có thể gây ra những hậu quả nghiêm trọng đối với các nhà đầu tư và người tiêu dùng.

Ý tưởng thực hiện mô hình dự đoán giá cổ phiếu của em

Bước 1: Thu thập dữ liệu giá cổ phiếu

Để đưa vào mô hình dự đoán giá cổ phiếu, ta cần thu thập dữ liệu giá cổ phiếu theo khoảng thời gian cố định. Ví dụ, ta có thể sử dụng dữ liệu giá cổ phiếu của một công ty trong 365 ngày gần nhất.

Bước 2: Chuẩn bị dữ liệu

Sau khi thu thập được dữ liệu giá cổ phiếu, ta cần tiến hành chuẩn bị dữ liệu cho mô hình. Đầu tiên, ta cần chia tập dữ liệu thành hai phần: tập huấn luyện và tập kiểm tra. Tập huấn luyện được sử dụng để huấn luyện mô hình, trong khi tập kiểm tra được sử dụng để kiểm tra hiệu quả của mô hình.

Tiếp theo, ta cần chuẩn hóa dữ liệu. Việc chuẩn hóa giúp đưa các giá trị dữ liệu về cùng một khoảng giá trị, giúp mô hình dễ dàng học và đưa ra dự đoán chính xác hơn. Có thể sử dụng phương pháp chuẩn hóa MinMaxScaler để chuẩn hóa dữ liệu.

Bước 3: Xây dựng mô hình GA-LSTM

Sau khi đã chuẩn bị dữ liệu, ta có thể bắt đầu xây dựng mô hình GA-LSTM để dự đoán giá cổ phiếu.

Thuật toán GA được sử dụng để tối ưu hóa các tham số của mô hình LSTM. Các tham số cần tối ưu bao gồm số lượng đơn vị của mỗi layer, số lượng layer, learning rate, số lượng epoch,...

Các bước cụ thể để xây dựng mô hình GA-LSTM như sau:

Khởi tạo các tham số của mô hình LSTM dưới dạng vector. Ví dụ, vector [100, 200, 0.001, 50] thể hiện số lượng đơn vị của layer 1, layer 2, learning rate và số lượng epoch tương ứng.

Sử dụng thuật toán GA để tối ưu hóa các tham số trên. Mỗi cá thể trong quần thể được biểu diễn bằng một vector tham số của mô hình. Các cá thể được đánh giá dựa trên giá trị hàm mục tiêu là MSE (Mean Squared Error) giữa giá dự đoán và giá thực tế.

Tiến hành huấn luyện mô hình LSTM với các tham số tối ưu được lấy từ quá trình tối ưu của thuật toán GA.

Sử dụng mô hình LSTM đã được huấn luyện để đưa ra dự đoán giá cổ phiếu trong tương lai.

Bước 4: Đánh giá mô hình

Sau khi đã xây dựng mô hình GA-LSTM, ta cần đánh giá hiệu quả của mô hình. Có thể sử dụng các độ đo như MSE, RMSE, MAE để đánh giá độ chính xác của mô hình.

Bước 5: Sử dụng mô hình để dự đoán giá cổ phiếu

Cuối cùng, ta có thể sử dụng mô hình GA-LSTM để dự đoán giá cổ phiếu trong tương lai. Với các giá trị đầu vào mới, ta có thể sử dụng mô hình đã huấn luyện để đưa ra dự đoán về giá cổ phiếu.

Ý tưởng cải thiện thuật toán của em

Sử dụng kỹ thuật ensemble learning: Kỹ thuật này kết hợp nhiều mô hình dự đoán khác nhau để đưa ra dự đoán cuối cùng. Việc kết hợp nhiều mô hình khác nhau có thể cải thiện độ chính xác và độ tin cậy của dự đoán.

Sử dụng các chỉ số kỹ thuật (technical indicators): Các chỉ số kỹ thuật được tính toán từ dữ liệu giá cổ phiếu như Moving Average, RSI, Bollinger Bands... Các chỉ số này có thể cung cấp thông tin hữu ích về xu hướng và biến động của giá cổ phiếu, giúp cải thiện độ chính xác của mô hình dự đoán.

Sử dụng dữ liệu bổ sung: Ngoài giá cổ phiếu, các thông tin khác như tin tức, sự kiện kinh tế, chính sách hay các thông tin về ngành công nghiệp, công ty cổ phiếu cũng có thể giúp cải thiện độ chính xác của mô hình dự đoán.

Tối ưu các tham số mô hình bằng các phương pháp khác: chẳng hạn như thuật toán Gradient Descent, hoặc các kỹ thuật tối ưu mô hình đang phát triển mới như Adam, RMSProp, Adagrad,...

***

Thuật toán Gradient Descent

Gradient Descent là một thuật toán tối ưu hóa được sử dụng để tìm giá trị tối ưu của một hàm số. Thuật toán này là một trong những phương pháp phổ biến nhất để tối ưu hóa các mô hình máy học, bao gồm các mô hình học sâu (deep learning).

Thuật toán Gradient Descent hoạt động bằng cách sử dụng đạo hàm của hàm số để tìm hướng di chuyển cho một điểm đến khi đạt được giá trị tối ưu của hàm số đó. Thuật toán bắt đầu bằng cách chọn một điểm bất kỳ trên đồ thị của hàm số và sau đó tính toán đạo hàm của hàm số tại điểm đó. Đạo hàm cho biết hướng tăng nhanh nhất của hàm số tại một điểm. Gradient Descent sau đó di chuyển điểm đó theo hướng ngược lại với đạo hàm để giảm giá trị của hàm số. Thuật toán tiếp tục thực hiện các bước này cho đến khi đạt được giá trị tối ưu của hàm số.

Thuật toán Gradient Descent có thể được sử dụng để tối ưu hóa các tham số trong mô hình máy học, bao gồm các tham số trong mô hình hồi quy tuyến tính, mạng nơ-ron, và các mô hình học sâu khác. Tuy nhiên, có thể tồn tại những trở ngại trong quá trình tối ưu hóa, ví dụ như điểm tối ưu cục bộ (local optimum) hoặc hàm số không đạt được giá trị tối ưu. Để giải quyết các vấn đề này, có thể sử dụng các biến thể của thuật toán Gradient Descent hoặc sử dụng các phương pháp tối ưu hóa khác.

***

Các mô hình Adam, RMSProp và Adagrad là các thuật toán tối ưu hóa trong Machine Learning để cập nhật trọng số của mạng neural trong quá trình huấn luyện.

Adam (Adaptive Moment Estimation): là một thuật toán tối ưu hóa kết hợp giữa hai thuật toán khác là Momentum và RMSProp. Adam cũng là một trong những thuật toán tối ưu phổ biến nhất hiện nay. Adam tính toán learning rate (tốc độ học) cho từng tham số trong quá trình huấn luyện, giúp tăng tốc độ hội tụ và giảm thiểu việc mắc phải tình trạng chậm hội tụ, đồng thời đảm bảo tính ổn định của thuật toán

RMSProp (Root Mean Square Propagation): là một thuật toán tối ưu hóa, được đề xuất bởi Geoffrey Hinton. Thuật toán này sử dụng moving average của bình phương gradient để điều chỉnh learning rate, giúp giảm thiểu vấn đề tăng learning rate quá nhanh và dẫn đến vấn đề overshooting.

Adagrad (Adaptive Gradient): là một thuật toán tối ưu hóa sử dụng learning rate tự điều chỉnh cho từng tham số. Thuật toán này tính toán adaptive learning rate bằng cách lưu trữ một bộ đếm cho mỗi tham số, đo lường tần suất xuất hiện của gradient tương ứng. Khi gradient xuất hiện thường xuyên, learning rate sẽ được giảm xuống để giảm thiểu độ nhạy của mô hình với các điểm dữ liệu lỗi ngoại lai…