BACKGROUND

QuantWater:
Traditional mineral waters offer fixed compositions that may not meet an individual's health requirements, leading to potential mineral imbalances. A personalized approach is needed to tailor mineral intake based on a comprehensive analysis of body fluid data.
HyperSign:
Existing document signing and form integration technologies (e.g., PDF-based systems like DocSign) are cumbersome, expensive, and do not integrate easily with enterprise databases. There is a need for a solution that is native to HTML and capable of seamless integration into modern ERP systems.
Adaptive Reweighting System:
Most machine learning models are trained offline on static datasets and lack mechanisms to differentiate between opinion and fact. With the prevalence of recycled or sensationalized content, a system that continuously adapts, verifies outputs against authoritative sources, and distinguishes fact from opinion is highly desirable.

SUMMARY OF THE INVENTION

QuantWater:
The invention provides a personalized mineralized water system comprising:

1. A body fluid collection module that obtains samples in a safe, hygienic manner.
2. An analysis module that precisely measures targeted minerals, vitamins, and bioactive compounds.
3. A computing module that calculates the optimal formulation for each individual.
4. A mineralization module that blends distilled water with the computed formulation.
5. A delivery module that ensures timely distribution of the personalized water product. Additionally, strategic partnerships with laboratories and healthcare providers facilitate ongoing monitoring and adjustments.

HyperSign:
The invention includes a computer-implemented method that enables users to drag and drop form elements (e.g., text fields, date pickers, signatures) directly into rendered HTML documents (from sources such as Microsoft Word). The system recognizes multiple target elements (TD, DIV, P, SPAN, etc.), adjusts display properties (inline or block), and automatically integrates these form fields with back-end databases for real-time Big Data analytics and workflow optimization.
Adaptive Reweighting System:
The invention provides a comprehensive framework for continuously updating a machine learning model's weights. Key components include:

1. A Data Acquisition Module that captures streaming data along with source credibility metadata.
2. A Preprocessing Module that tokenizes and extracts features from the data.

3. An Adaptive Reweighting Module that uses online optimization (e.g., gradient descent) to update weights in real time while mitigating catastrophic forgetting via regularization techniques.
4. A Content Differentiation Module that distinguishes between opinion-based content and verifiable, action-backed statements using NLP techniques and factuality scoring.
5. An Audit Verification Module that compares outputs with trusted references, triggering corrective feedback if discrepancies arise.
6. A Monitoring and Control Loop that continuously evaluates performance metrics and triggers further updates if necessary.
7. An Update and Deployment Engine that seamlessly integrates new weights into the live system.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 – Flow Diagram for QuantWater:
A flowchart illustrating the sequential steps from body fluid collection, analysis, personalized formulation, mineralization, and delivery, with integrated partnerships for ongoing monitoring.

```
Flowchart TD

    A[Body Fluid Collection Module]

    B[Analysis Module<br/>(Mineral, Vitamin, & Bioactive Measurement)]

    C[Computing Module<br/>(Personalized Formulation Calculator)]

    D[Mineralization Module<br/>(Precise Blending with Distilled Water)]

    E[Delivery Module<br/>(Direct Distribution to Customer)]

    F[Partner Laboratories & Healthcare Providers]



    A --> B

    B --> C

    C --> D
```

D --> E

F --- B

F --- C

Figure 2 – Flow Diagram for HyperSign:
A conceptual diagram showing a source document (e.g., Microsoft Word) rendered into HTML, with draggable form elements being dynamically placed into target HTML elements and integrated with enterprise systems.

flowchart TD

A [Source Document
(e.g., Microsoft Word)]

B [HTML Rendering Engine]

C [Draggable Elements
(Form Fields: Text, Date Picker, Signature, etc.)]

D [Target HTML Elements
(TD, DIV, P, SPAN, etc.)]

E [Integration Module
(Dynamic Form Placement & Styling)]

F [Enterprise Systems & ERP
(Database Integration & Big Data Analytics)]

A --> B

B --> D

C --> E

D --> E

E --> F

Figure 3 – Flow Diagram for the Adaptive Reweighting System:
A diagram depicting the flow of data through the Data Acquisition Module, Preprocessing Module, Content Differentiation Module, Adaptive Reweighting Module, Audit Verification Module, and Update & Deployment Engine.

flowchart TD

A[Data Acquisition Module
(Capture streaming data with credibility metadata)]

B[Preprocessing Module
(Normalize, tokenize, and extract features)]

C[Adaptive Reweighting Module
(Online optimization & weight update)]

D[Content Differentiation Module
(Distinguish fact-based vs opinion-based content
Assign factuality scores)]

E[Audit Verification Module
(Cross-check outputs against trusted references)]

F[Monitoring and Control Loop
(Evaluate performance metrics
Trigger reweighting when necessary)]

G[Update and Deployment Engine
(Integrate updated weights into live system)]

A --> B

B --> C

C --> D

D --> E

E --> F

F --> G

%% Optionally, feedback loop from G back to A or C

G -- Feedback --> C

DETAILED DESCRIPTION OF THE INVENTION

QuantWater: Personalized Mineralized Water System

1. **Body Fluid Collection Module:**
   Describes the safe, hygienic method for obtaining body fluid samples.
2. **Analysis Module:**
   Details the specialized analyzers for precise measurement of minerals, vitamins, and bioactive compounds.
3. **Computing Module:**
   Explains the algorithm used to calculate the personalized formulation.
4. **Mineralization Module:**
   Describes the apparatus for accurately blending the computed formulation with distilled water while ensuring contaminant removal.
5. **Delivery Module:**
   Outlines the logistics and integration with delivery services to provide the personalized water directly to the customer.
6. **Partnership Integration:**
   Discusses strategic partnerships with laboratories and healthcare providers for free blood analysis, referral networks, and ongoing monitoring.

HyperSign: Native HTML Form Integration Protocol

1. **Source Document Processing:**
   Describes the conversion of a Microsoft Word document into HTML.
2. **Draggable Elements:**
   Details the interactive components (text fields, date pickers, signatures, etc.) that can be dynamically positioned.
3. **Target HTML Recognition:**
   Explains the algorithm that identifies valid drop targets (e.g., TD, DIV, P, SPAN) and adjusts element styling (inline or block).
4. **Database Integration:**
   Covers how the dynamically created form fields automatically integrate with backend systems for Big Data analytics and ERP integration.
5. **Workflow Efficiency:**
   Outlines how this system disrupts traditional PDF-based signing by reducing software development costs and speeding up implementation for regulatory, compliance, and operational processes.

Adaptive Reweighting System for Machine Learning Models

1. **Data Acquisition Module:**
   Captures streaming data along with metadata indicating source credibility.
2. **Preprocessing Module:**
   Normalizes and tokenizes data; extracts relevant features.
3. **Adaptive Reweighting Module:**
   Utilizes online optimization (e.g., online gradient descent, adaptive moment estimation) and incorporates regularization techniques to prevent catastrophic forgetting.
4. **Content Differentiation Module:**
   Analyzes textual inputs to distinguish opinion-based content from factual, action-backed statements via NLP and assigns factuality scores.
5. **Audit Verification Module:**
   Cross-checks outputs against trusted, authoritative sources and provides feedback for corrective weight updates.
6. **Monitoring and Control Loop:**
   Continuously evaluates performance metrics (accuracy, precision, recall, loss) and triggers reweighting when deviations occur.
7. **Update and Deployment Engine:**
   Seamlessly integrates updated weights into the live system with minimal downtime, using version control and scheduled deployment strategies.

SUPPLEMENTARY DISCLOSURE

A system and method for continuously reweighting the parameters of a machine learning model with integrated audit verification and content differentiation is disclosed. The invention provides an architecture for online adaptation whereby model weights are incrementally updated using incoming data streams, and outputs are continuously compared against authoritative sources to validate factual accuracy. A dedicated content differentiation module parses incoming information to distinguish between opinions and verifiable statements based on actual actions. This ensures that the system prioritizes data backed by objective evidence over recycled or subjective content. The overall system includes data acquisition, preprocessing, an adaptive reweighting module, an audit verification module, a content differentiation module, and a monitoring/control loop that dynamically adjusts weight parameters in response to both performance metrics and external fact-based validations.

Background: Modern machine learning models, particularly large-scale neural networks, are generally trained offline on fixed datasets and do not inherently distinguish between objective facts and subjective opinions. Given the prevalence of media content that often recycles opinions rather than factual, action-backed statements, there is an increasing need for systems that can:

Continuously adapt to new data, Validate outputs against verified facts, And, crucially, parse the difference between opinions and statements backed by actual actions.

Traditional transformer models capture statistical correlations from training data without explicit mechanisms to filter out sensationalized or opinion-based content. There is thus a critical need for a system that continuously reweights itself, integrates external fact verification, and distinguishes between subjective opinions and verifiable facts.

Summary: The present invention provides a system and method for continuously reweighting machine learning model parameters with integrated audit verification and content differentiation. In one embodiment, the system comprises:

Data Acquisition Module: Captures streaming data from multiple sources, including metadata about source credibility.

Preprocessing Module: Normalizes, tokenizes, and converts incoming data into structured feature representations.

Adaptive Reweighting Module: Employs online optimization techniques to update model weights in real time, incorporating mechanisms to prevent catastrophic forgetting.

Content Differentiation Module: Analyzes incoming information to distinguish between subjective opinions and statements that are backed by verifiable actions. This module leverages natural language processing techniques and external reference checks to assign a factuality score to content.

Audit Verification Module: Cross-checks generated outputs against trusted reference repositories (e.g., verified regulatory texts) to ensure accuracy.

Monitoring and Control Loop: Continuously evaluates model performance using predefined metrics and feedback from the audit and content differentiation modules, triggering adaptive reweighting as needed.

Update and Deployment Engine: Seamlessly deploys updated model weights into the live system with minimal downtime.

This enhanced architecture ensures that the system not only continuously updates its internal representations with new data but also filters out sensationalized or recycled opinions in favor of factual, action-backed statements.

Detailed Description

System Architecture

1. Data Acquisition Module

Function: Continuously collects streaming data from diverse real-time sources, such as news feeds, regulatory databases, social media, and sensor networks.

Components: Interfaces for data ingestion, buffering mechanisms, and metadata extraction (including source credibility indicators).

2. Preprocessing Module

Function: Transforms raw data into structured inputs that are compatible with the model.

Components:

Tokenizer/Normalizer: Converts raw text into tokens and standardizes formats.

Feature Extractor: Identifies salient features from text and associated metadata.

Data Validator: Checks data integrity and alignment with the model's input specifications.

3. Adaptive Reweighting Module

Function: Continuously updates the model weights using online optimization algorithms.

Components:

Online Optimizer: Implements online gradient descent, adaptive moment estimation, or other suitable algorithms to compute gradients based on new data.

Regularization Mechanism: Incorporates memory buffers or dual-memory architectures to mitigate catastrophic forgetting.

Weight Update Engine: Applies calculated weight adjustments in real time.

4. Content Differentiation Module

Function: Analyzes incoming content to distinguish between opinion-based material and statements grounded in verifiable actions.

Components:

Opinion Detection: Uses linguistic cues (e.g., hedging language, sentiment markers, and subjectivity indicators) to flag opinionated content.

Fact-Based Analysis: Cross-references statements with external verified databases or reference repositories to identify verifiable actions and factual events.

Factuality Scoring: Assigns scores to data inputs based on their degree of objectivity and support from authoritative sources, thereby influencing their weight during training.

Operation: For example, when processing media content about a public figure, the module distinguishes between sensational headlines (e.g., claims based on a freeze-frame image) and reports corroborated by multiple, verifiable sources detailing actual actions. Data that scores high on factuality is given greater influence during weight updates.

5. Audit Verification Module

Function: Validates the model's outputs by comparing them against trusted external references.

Components:

Reference Comparator: Accesses a controlled repository of authoritative data (e.g., verified FDA guidelines or historical political records).

Discrepancy Detector: Identifies deviations between generated outputs and reference information.

Feedback Integrator: Provides corrective feedback to the Adaptive Reweighting Module if discrepancies are detected.

6. Monitoring and Control Loop

Function: Continuously monitors model performance and the accuracy of its outputs.

Components:

Performance Evaluator: Assesses outputs using metrics such as accuracy, precision, recall, and loss.

Threshold Controller: Sets acceptable performance boundaries and triggers reweighting when deviations occur.

Feedback Interface: Integrates insights from the Audit Verification and Content Differentiation Modules to guide adaptive learning.

7. Update and Deployment Engine

Function: Seamlessly deploys updated model weights into the live system.

Components:

Version Manager: Maintains version control and facilitates rollback if necessary.

Deployment Scheduler: Plans optimal update times to minimize service disruption.

Validation Subsystem: Runs final tests against curated validation datasets before full deployment.

Flow Diagram (Conceptual)

Data Input:

Streaming data with associated metadata → Data Acquisition Module

Preprocessing:

Raw data → Tokenizer/Normalizer → Feature Extraction

Content Differentiation:

Preprocessed data → Content Differentiation Module (assigns factuality scores; flags opinions)

Weight Update Cycle:

Fact-verified data + Current Model → Adaptive Reweighting Module → Compute Gradients → Update Weights

Audit Verification:

Generated outputs → Audit Verification Module → Compare with Trusted References → Feedback

Monitoring:

Updated Model → Performance Evaluator & Discrepancy Detector → Feedback to Control Loop

Deployment:

Validated Model → Update and Deployment Engine → Live System

Example Embodiment

Consider a neural network deployed to provide political analysis in real time. Given the abundance of sensationalized headlines, the system operates as follows:

Data Collection: Continuously ingests news articles, social media posts, and official statements.

Content Differentiation: The module distinguishes between opinion-based content and verifiable actions (such as confirmed visits, public statements, or documented events).

Adaptive Reweighting: The system updates its weights by giving more prominence to data with high factuality scores.

Audit Verification: When the model generates outputs (e.g., statements regarding a public figure's actions), these outputs are compared against a repository of verified records. Discrepancies trigger corrective feedback.

Live Adaptation: The entire cycle ensures that the model adapts not only to new data but does so by filtering out opinions in favor of objective, fact-backed statements.

A system and method for continuously reweighting the parameters of a machine learning model is disclosed. The invention provides an architecture for online adaptation whereby model weights are incrementally updated using incoming data streams. The system includes data acquisition, preprocessing, an adaptive reweighting module, and a monitoring/control loop that dynamically adjusts weight parameters in response to performance metrics. This continuous learning approach mitigates issues such as model staleness and catastrophic forgetting, thereby ensuring that the model remains current with evolving data distributions.

Background: Modern machine learning models, particularly large-scale neural networks, are typically trained offline on fixed datasets. Once deployed, these models operate with static parameters that do not change in response to new data. However, the rapid evolution of data domains and the need for models to adapt to changing environments have underscored the limitations of fixed-weight systems.

Current methods for online or continuous learning face challenges including:

- *Catastrophic Forgetting: The tendency for new data to override previously learned information.*

- *Stability-Plasticity Trade-Off: Balancing the retention of historical knowledge while integrating new patterns.*

- *Computational Overhead: Real-time updating can be computationally intensive and may compromise model performance.*

There is a need for a system that continuously reweights model parameters in a controlled manner, ensuring that the model remains both stable and adaptable over time.

Summary: The present invention provides a system and method for continuously reweighting machine learning model parameters. In one embodiment, the system comprises:

1. *Data Acquisition Module: Captures streaming data from one or more sources.*

2. *Preprocessing Module: Normalizes and tokenizes incoming data to align with the model's input format.*

3. *Adaptive Reweighting Module: Uses online optimization techniques to update model weights in real time. This module may incorporate mechanisms to prevent catastrophic forgetting, such as memory replay buffers, regularization terms, or dual-memory architectures.*

4. *Monitoring and Control Loop: Continuously assesses model performance against pre-defined metrics. When performance deviates beyond a threshold, the control loop triggers the reweighting process to update the model.*

5. *Update and Deployment Engine: Applies the new weight parameters in a manner that minimizes downtime and ensures smooth transition from previous model states.*

This approach allows the system to function as a continuously learning neural network that can adjust its internal representations and predictions based on the latest available data.

Detailed Description: System Architecture

1. Data Acquisition Module

- *Function: Collects data from various real-time sources (e.g., IoT devices, web streams, sensors).*

- *Components: Interfaces for data ingestion, buffering mechanisms to handle variable data rates, and preliminary filtering to remove noise.*

2. Preprocessing Module

- *Function: Transforms raw data into a structured format suitable for the model.*

- *Components:*

  - *Tokenizer/Normalizer: Converts input data into tokens or embeddings.*

  - *Feature Extractor: Identifies salient features for use by the reweighting module.*

  - *Data Validator: Ensures data integrity and compatibility with the model's expected input.*

3. Adaptive Reweighting Module

- *Function: Performs continuous updates on model weights using the latest data.*
- *Components:*

o *Online Optimizer: Utilizes algorithms (e.g., online gradient descent, adaptive moment estimation) that calculate gradients on incoming data batches.*

o *Regularization Mechanism: Prevents overfitting and catastrophic forgetting by applying constraints or leveraging historical data stored in a memory buffer.*

o *Weight Update Engine: Applies computed weight adjustments to the model's parameters in a controlled manner.*

- *Operation:*

When new data arrives, the module computes the gradient of a loss function with respect to current weights. This gradient is then used to update the weights incrementally. A regularization term may be included to ensure that these updates do not lead to a complete override of previously learned knowledge.

4. Monitoring and Control Loop

- *Function: Continuously monitors the model's performance metrics (e.g., accuracy, precision, recall, loss).*
- *Components:*

o *Performance Evaluator: Compares model predictions against ground truth or reference standards.*

o *Threshold Controller: Defines acceptable performance ranges and triggers reweighting processes if metrics fall outside these ranges.*

o *Feedback Interface: Provides feedback to the adaptive reweighting module to adjust learning rates or update frequency.*

5. Update and Deployment Engine

- *Function: Integrates newly updated weights into the live model with minimal disruption.*
- *Components:*

o *Version Manager: Maintains versions of model weights and facilitates rollback if necessary.*

o Deployment Scheduler: Determines optimal times for updating the live model to ensure continuous service.

o Testing Subsystem: Validates the updated model on a separate validation dataset before full deployment.

Flow Diagram (Conceptual)

1. Data Input:

Real-time data → Data Acquisition Module

2. Preprocessing:

Raw data → Tokenizer/Normalizer → Feature Extractor

3. Weight Update Cycle:

Preprocessed data + Current Model → Adaptive Reweighting Module → Compute Gradients → Update Weights

4. Monitoring:

Updated Model → Performance Evaluator → Feedback to Control Loop

5. Deployment:

Validated Model → Update and Deployment Engine → Live System

Example Embodiment

Consider a deployed neural network model for real-time language translation. As new colloquial expressions and slang emerge, the model's performance might degrade. Using the system described:

- *New linguistic data is captured by the Data Acquisition Module.*
- *The Preprocessing Module tokenizes and extracts features from this language data.*
- *The Adaptive Reweighting Module computes adjustments to the language model's weights based on these new expressions.*
- *The Monitoring and Control Loop checks the updated model's translation accuracy.*
- *If performance improves and meets thresholds, the updated weights are deployed seamlessly, ensuring the translation model remains current with evolving language usage.*