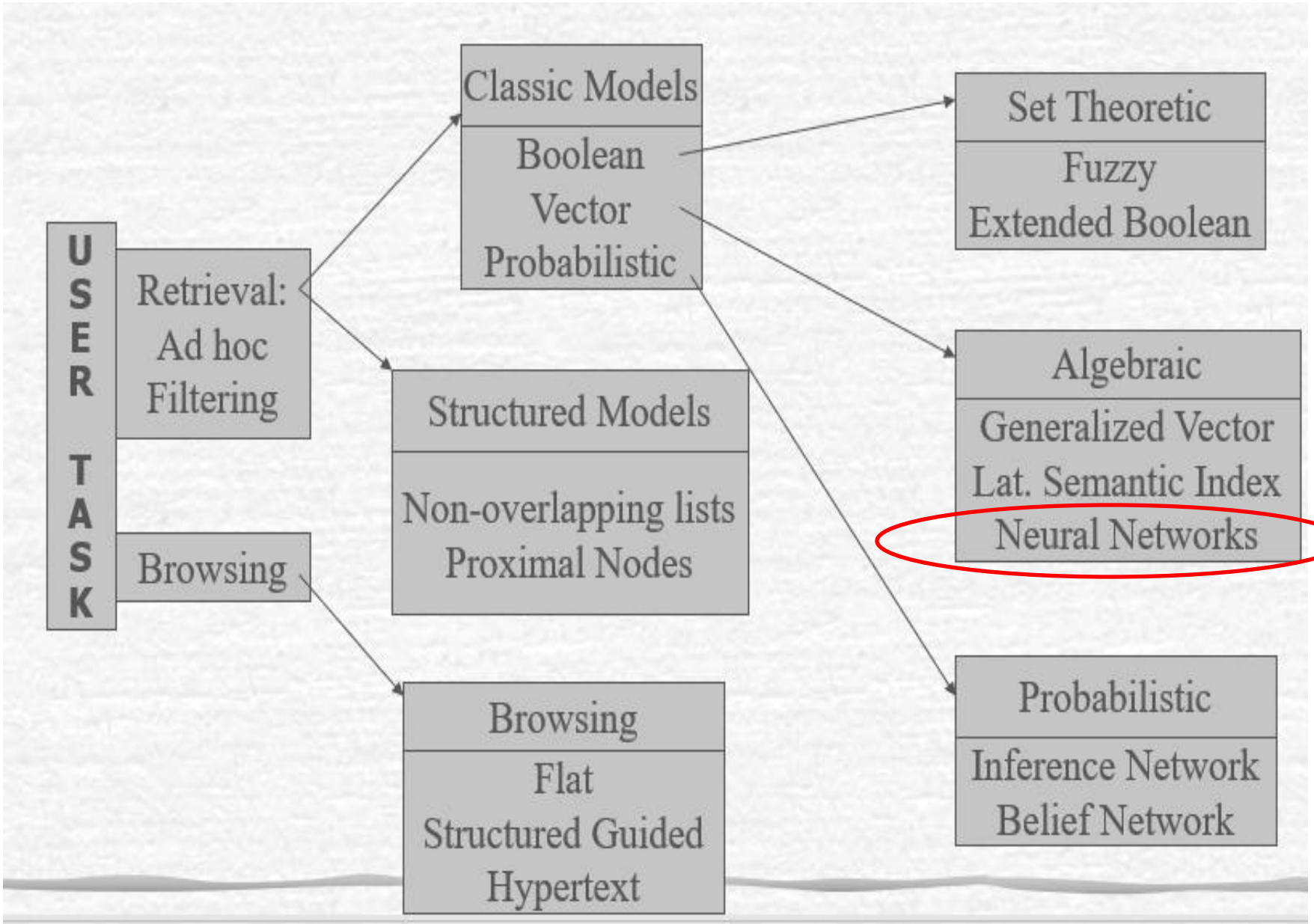
Information Retrieval

By
Dr Syed Khaldoon Khurshid

Presentation for Tomorrow

- Five Topics Presentation is due tomorrow
- Each section will make a single presentation
- Topics and groups are being allocated.
- Marks Range 10-50. Absent will get -10 marks.
- Today absentees must join already created groups and present tomorrow.
- Maximum time for each group is 10 minutes.
- Both Projects and computed example will be used for the section presentation, with the focus on the project 1.
- Use Richard Feynman's technique for the preparation and presentation of the slides.


USER TASK
Retrieval: Ad hoc Filtering
Browsing
Classic Models
Boolean
Vector
Probabilistic
Structured Models
Non-overlapping lists
Proximal Nodes
Browsing
Flat
Structured Guided
Hypertext
Set Theoretic
Fuzzy
Extended Boolean
Algebraic
Generalized Vector
Lat. Semantic Index
Neural Networks
Probabilistic
Inference Network
Belief Network

Lecture of the week

- **Algebraic**
  - **Neural Networks**

Set theoretic Information Retrieval (IR) models: Neural Networks

- Set theoretic Information Retrieval (IR) models, including those based on **neural networks**, are ***extensions of classical IR models*** that offer more advanced and accurate ways to retrieve information. Here's a simple concept to understand how they work:

Traditional IR Models (Classical):

- In traditional IR models, when you search for something, the system looks for **exact matches of your keywords** in documents. If your query contains the word "apple," it finds documents with the word "**apple**" in them.

Set theoretic Information Retrieval (IR) models: Neural Networks

Set Theoretic IR Models (Extension):

- Set theoretic models, including neural networks, go beyond exact matches. **They consider the relationships between words and documents**.

Think of it like this:

- **Words as Sets**: Imagine each word in a document is like a set of related words. For example, the word "apple" is related to "fruit," "orchard," and "healthy."
- **Documents as Sets of Words:** Documents are collections of these sets (words). Each document contains sets of words related to its content.

Set theoretic Information Retrieval (IR) models: Neural Networks

- **Neural Networks:** Neural networks, like the human brain, learn these relationships. **They understand that "apple" is related to "fruit," and "fruit" is related to "healthy."** So, if you search for "**healthy snacks**," it can find documents talking about apples as a healthy snack, even if they don't use the exact phrase.
- **Improved Relevance:** Set theoretic models, powered by neural networks, **help find documents that are more relevant to your query by understanding the connections between words**. They consider not only the words you use but also their relationships, making information retrieval smarter and more accurate.

Weights Vs Biases

- Let's consider a simple information retrieval (IR) example in the context of weights and biases:
- **Scenario:** Imagine you have a search engine that retrieves articles about **healthy eating based on user queries**. The goal is to **rank articles according to their relevance to the query**.
- **Weights:** In this case, weights would represent **the importance assigned to different words or terms within the articles**. For instance, the word "apple" might be given a higher weight because it's a strong indicator of a healthy snack. The network learns through training that "apple" is an important keyword for relevance.

Weights Vs Biases

- **Biases:** Biases could be thought of as **a baseline relevance score for each article**. They represent the likelihood that ***an article is relevant to healthy eating, even if none of the specific keywords match the query exactly.*** **This baseline score ensures that even articles without exact query terms can still be considered relevant.**

- So, the **weights** determine **the importance of specific words in articles**, and **biases** **account for the overall relevance of articles**, even if they don't perfectly match the query. This **combination helps the IR system rank and retrieve articles** effectively.

An Approach to computational modeling of Neural Networks

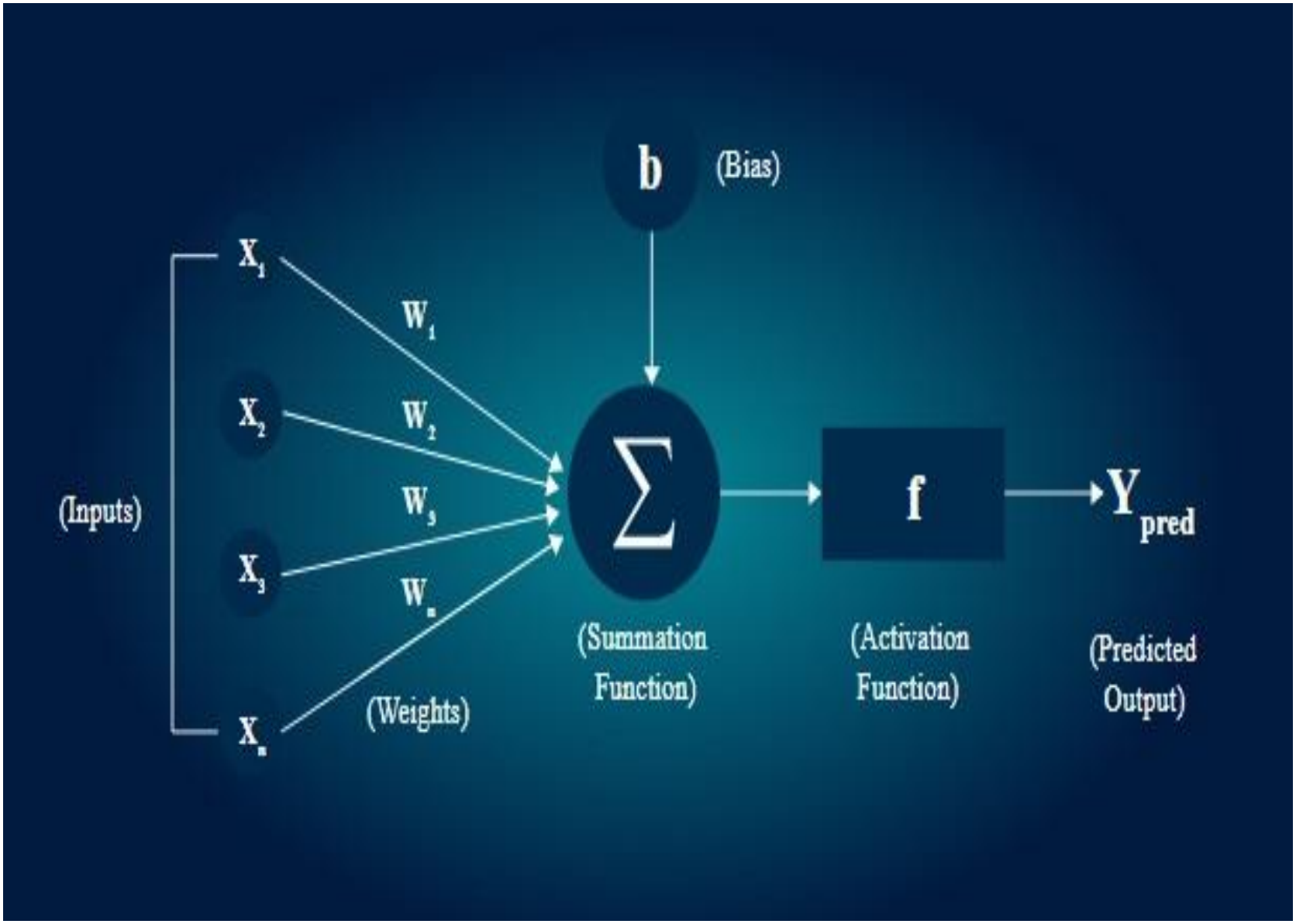
A step-by-step approach to computational modeling of Neural Networks for information retrieval:

1. **Data Preparation:** Collect and preprocess text data, which includes cleaning, tokenization, and encoding.
2. **Data Representation:** Convert text into numerical vectors, often using techniques like Word Embeddings (e.g., Word2Vec or GloVe).
3. **Architecture Design:** Define the neural network architecture, including the type of layers (e.g., input, hidden, and output layers), activation functions, and the number of neurons in each layer.
4. **Training Data:** Split the dataset into a training set and a validation set. The training set is used to train the neural network, and the validation set is used to tune hyperparameters.

5. **Loss Function:** Choose an appropriate loss function that quantifies the difference between predicted and actual outcomes. In information retrieval, common loss functions include ***categorical cross-entropy or mean squared error***.
6. **Back propagation:** Train the neural network using optimization algorithms (e.g., stochastic gradient descent) to minimize the loss function. This involves iteratively adjusting the model's weights and biases.
7. **Validation:** Monitor the model's performance on the validation set during training to prevent over fitting.
8. **Testing:** Evaluate the model on a separate test dataset to assess its real-world performance.
9. **Inference:** Once the model is trained and tested, you can use it to make predictions on new, unseen data.
10. **Semantic Understanding:** Neural networks capture semantic relationships between words, allowing them to find relevant documents based on context and meaning, even if specific keywords or phrases are not used.

- **By following these steps, neural networks can learn complex relationships in text data, enabling more accurate information retrieval and understanding of user intent.**

Neural Network



Projectof information retrieval using a neural network

A simplified real-world example of information retrieval using a neural network:

Step 1: Data Preparation

- Imagine we have a **dataset of customer reviews for a product**. Each review is in the form of text. We preprocess this data by cleaning it, removing punctuation, and splitting it into individual words or tokens.

Step 2: Data Representation

- Next, we convert the text into numerical vectors using Word Embeddings. For instance, we represent **each word as a high-dimensional vector where similar words are closer in the vector space**. This allows the neural network to understand semantic relationships between words.

Step 3: Architecture Design

- We design a neural network with an input layer, one or more hidden layers, and an output layer. **The hidden layers use activation functions like ReLU to model complex patterns in the data.**

Step 4: Training Data

- **We split the dataset into a training set** and a validation set. The training set contains customer reviews and their corresponding labels, such as "positive" or "negative" sentiment. The validation set will be a **separate subset of the dataset**. It will also contain customer reviews along with their labels (either "positive" or "negative").

Step 5: Loss Function

- We choose a loss function that quantifies the **difference between the predicted sentiment (output) and the actual sentiment (label).** For sentiment analysis, a common loss function is **categorical cross-entropy.**

Step 6: Back propagation

- We train **the neural network using an optimization algorithm** like **stochastic gradient descent**. The model iteratively adjusts its weights and biases to minimize the loss function.

Step 7: Validation

- During training, **we monitor the model's performance on the validation set to ensure it generalizes well to new data**.
- The key purpose of the **validation set is to check how well the model generalizes to unseen data during training**. After training on the training set, the model is evaluated on the validation set to track its performance and tune hyper-parameters, **such as learning rate or regularization parameters**. This way, the model is not only optimized for the training data but also performs well on new, unseen data.

Step 8: Testing

- We evaluate the trained model on **a separate test dataset** to measure its accuracy in predicting sentiment.

Step 9: Inference

- Now, we can use the trained neural network **to analyze new customer reviews**. It can predict whether a review is positive or negative based on the context and meaning of the words used, even if the exact phrases from the training data are not present.

Project-II of a Neural Network

- A simple project of how a neural network can be applied to text data in the context of Information Retrieval (IR) using a common daily scenario:

Scenario: Voice Assistant

- Imagine you're using a voice-activated assistant like Siri, Google Assistant, or Alexa to find information online. You say, "**Find me articles about the benefits of exercise for health**."

1. **Speech-to-Text Conversion:** Your spoken query is first converted to text by the voice assistant**. For example**, it transcribes your request as "Find me articles about the benefits of exercise for health.“

Project-II of a Neural Network

2. **Text Preprocessing:** The neural network then processes this text. It breaks down the query into its components, such as keywords and context. In this case, it recognizes keywords like "benefits," "exercise," and "health."
3. **Semantic Understanding:** The neural network understands the meaning behind your words. It knows that "benefits" refers to advantages, "exercise" is related to physical activity, and "health" is about well-being.
4. **IR Model Interaction:** The neural network interacts with an IR model. This model uses its training data and understands that your query is not just about exact word matches. It should also consider **synonyms and related terms**. For example, "benefits" could be related to "**advantages**" "exercise" can include "**workouts**" and "health" might encompass "**wellness**."

Project-II of a Neural Network

5. **Query Expansion:** The neural network **expands your query by considering these synonyms and related terms**. It searches for articles that match not only the exact keywords but also the expanded terms.
6. **Search and Retrieval:** The network sends this refined query to a search engine. The search engine retrieves articles and documents that best match the expanded query. **It considers the semantic understanding provided by the neural network.**
7. **Response:** The voice assistant reads out the relevant articles to you, such as "Here are some articles about the benefits of regular workouts for your overall well-being."

A simplified mathematical expression for a Feed Forward Neural Network

- A simplified mathematical expression for a feed forward neural network commonly used in Information Retrieval (IR) with AI:
- **Let's consider a neural network with a single hidden layer. The input to the network is a feature vector X, and the output is the relevance score for a document.**

1. **Input Layer:** The input layer consists of features extracted from a document-query pair, represented as a feature vector X.
2. **Hidden Layer:** In this single hidden layer, we **calculate the weighted sum of the input features**, apply an activation function (commonly the **sigmoid or ReLU function**), and produce the hidden layer's output.

A simplified mathematical expression for a feed forward Neural Network

Let's represent this as H

H = Activation(W_input * X + B_input)

W_input: Weight matrix for the input features.

B_input: Bias term for the hidden layer.

Activation: The activation function (e.g., sigmoid or ReLU).

3. **Output Layer:** The hidden layer's output is then used to calculate the relevance score for the document:

   Relevance Score = W_output * H + B_output

   W_output: Weight matrix for the output layer.

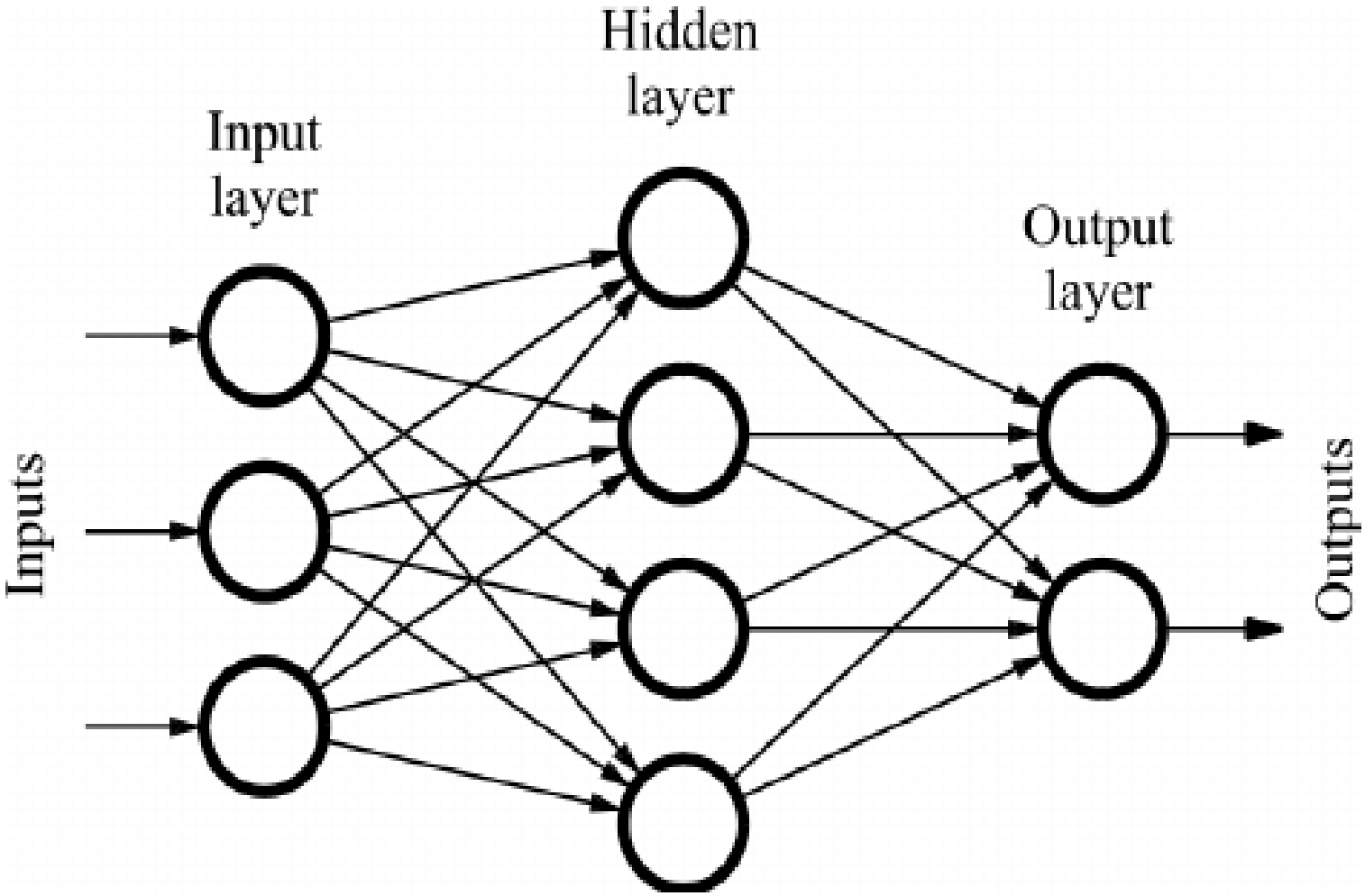
   B_output: Bias term for the output layer.

4. **Activation Function:** The activation function in the hidden layer introduces non-linearity and allows the network to model complex relationships between input features.

A simplified mathematical expression for a feed forward Neural Network

- T**he network is trained using labeled data** (document-query pairs with relevance scores) **to learn the optimal values for the weights and biases** (W_input, B_input, W_output, B_output) **that minimize the prediction error.**
- In practice, neural networks used in IR are often more complex, with multiple hidden layers and various optimization techniques. However, this simplified expression represents the **basic structure of a feed forward neural network for IR.**
- **The goal is to learn a mapping from input features to relevance scores, allowing the network to rank documents based on their relevance to a given query.**

Feed Forward Neural Network



Understanding the example

- In the example of the neural network for Information Retrieval, we have a system that helps rank documents based on their relevance to a given query. Let's break down the key concepts:

1. **Input Features (X):** These are characteristics or attributes extracted from documents and queries. In our example, we have two features, **X1 and X2**, with numerical values representing some document-query relationship.

Understanding the example

2. **Weights (W_input):** Weights are parameters that the network learns during training. They signify the importance of each input feature. In our example, W_input1 and W_input2 are weights associated with X1 and X2, respectively.

3. **Bias (B_input):** The bias term allows the network to introduce some flexibility and adjust its predictions. It's like an extra knob in the system. In our example, B_input is the bias for the hidden layer.

- ***The bias term (B input) in a neural network adjusts the output independently of the input features, providing the network with the flexibility to shift activation thresholds for better learning and fitting the data.***

Computed Example

Let's populate the simplified neural network expression with numerical values for a hypothetical example in Information Retrieval:

Suppose we have the following values:

Input Features (X): A feature vector with two components, X1 and X2.

- X1 = 0.7
- X2 = 0.5

Weights (W_input) for the hidden layer:

- W_input1 = 0.8
- W_input2 = 0.6

Bias (B_input) for the hidden layer:

- B_input = 0.2

Computed Example

Weights (W_output) for the output layer:

- W_output = 1.5

Bias (B_output) for the output layer:

- B_output = -0.4

Activation Function (Sigmoid) for the hidden layer:

Let's calculate the relevance score using the neural network formula:

Hidden Layer Output (H):

- H = Sigmoid(0.8 * 0.7 + 0.6 * 0.5 + 0.2)
- H = Sigmoid(0.56 + 0.3 + 0.2)

Understanding the example

5. **Sigmoid Activation:** The sigmoid function squashes the weighted sum of inputs into a range between 0 and 1, representing the probability that the document is relevant. In our example, if H is approximately 0.742, indicating a certain level of relevance.

6. **Weights (W_output) and Bias (B_output):** These are parameters specific to the output layer. They are used to map the hidden layer's output (H) to the final relevance score. In our example, W_output and B_output help calculate the relevance score.

Computed Example

Now, let's calculate the relevance score using the neural network formula:

Hidden Layer Output (H):

- H = Sigmoid(0.8 * 0.7 + 0.6 * 0.5 + 0.2)
- H = Sigmoid(0.56 + 0.3 + 0.2)
- H = Sigmoid(1.06)

Assuming a typical sigmoid function, the output (H) after applying the sigmoid activation might be approximately 0.742.

Relevance Score (Output):

Relevance Score = 1.5 * 0.742 - 0.4

Relevance Score ≈ 1.113 - 0.4

Relevance Score ≈ 0.713

Understanding the example

7. **Relevance Score:** This is the ultimate output of the neural network, indicating how relevant a document is to the query. In our example, the relevance score is approximately 0.713, suggesting the document is moderately relevant.

- In practice, **neural networks like this one are trained on large datasets with labeled examples to learn the most appropriate weights and biases.** The network's goal is to predict relevance scores that match human judgments about document relevance, improving the accuracy of document ranking in Information Retrieval tasks.

Computed Example

- In this example, the neural network takes input features **X1 = 0.7 and X2 = 0.5**, processes them through the **hidden layer with specific weights and biases**, **applies the sigmoid activation function**, and **produces a relevance score of approximately 0.713**.

- **This simplified example demonstrates the basic calculation within a neural network used in Information Retrieval, where the network learns to assign relevance scores to documents based on input features**.