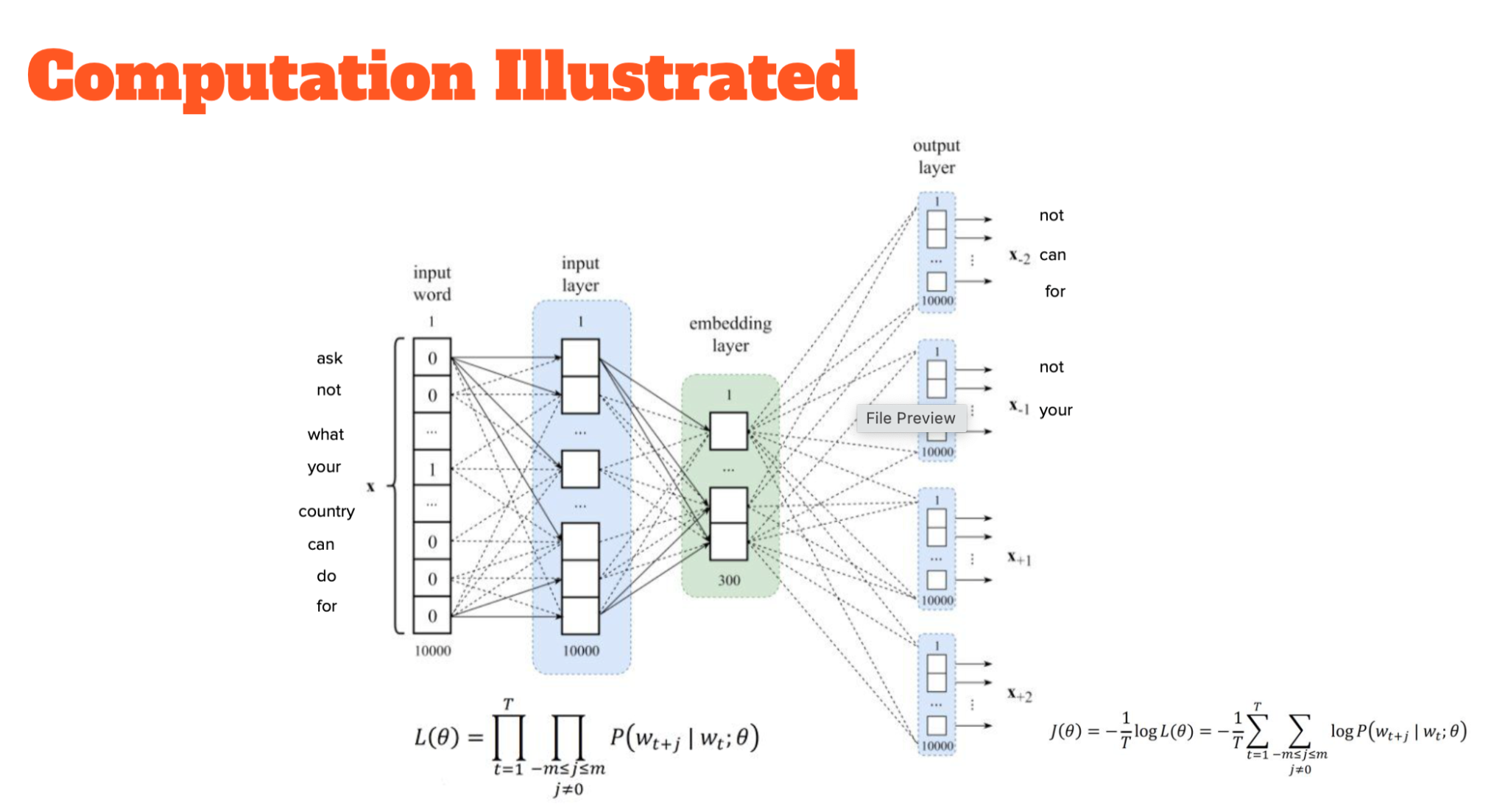
**Understanding Continuous Bag of Words (COBW)**



*(The above image is from Dr. Begoli’s lecture 4)*

**Input Layer:**

Each word is encoded as a **one-hot vector**, meaning only one position in the vector is set to "1," representing the word, while all other positions are "0." For example, in a vocabulary of 1000 words, the word "amazing" might be represented as a 1000-dimensional vector, with a "1" in the position corresponding to "amazing" and "0" elsewhere.

In the sentence:  
*"The goal: to determine who must call the other an amazing detective/genius,"*We have to predict the word *"detective"* using the context words *"amazing"* and *"genius."* These context words are encoded as one-hot vectors and fed into the model as input.

**Embedding Layer:**

The **embedding layer** transforms the high-dimensional one-hot encoded input vectors into lower-dimensional, dense vectors called **word embeddings**. These embeddings are vectors that represent the semantic meaning of words. Initially, these vectors are randomly assigned, but through training, the model adjusts them to capture relationships between words.

For example, the word *"amazing"* might initially be represented as a 300-dimensional vector of random values. After training, this vector will change based on how *"amazing"* is used in context, becoming more meaningful and reflecting its relationship to other words like "*genius."*

**Output Layer:**

The output layer predicts the **target word** (in this case, *"detective"*) based on the embeddings of the context words (*"amazing"* and *"genius"*). The model takes the embeddings from the hidden layer, combines them, and computes the probability distribution over all possible words in the vocabulary. It then predicts the word that is most likely to fit the context.

This can be thought of as the model asking, "Given the words 'amazing' and 'genius,' which word makes the most sense as the target word in this context?"

**Loss Function:**

At the bottom of the image, the **loss function** is described mathematically. The **likelihood function L(θ)** represents the probability of predicting the correct target word given the context words and model parameters θ(the weights of the neural network).

The **loss function J(θ)** is the **negative log-likelihood** (or **cross-entropy loss**), which measures how far the model’s predictions are from the actual target word. The model aims to minimize this loss by adjusting the embeddings and weights, so the predicted word becomes more accurate.

* T: Total number of words in the text (the training corpus).
* Wt: Target word
* Wt+j: Context words surrounding wt, where j ranges from -m to m
* P(wt+j | wt; θ): Probability of predicting the context word wt+j given the target word wt, where θ represents the model parameters (the weights and biases of the neural network).