Neural Network CV Prep

BackPropagation:

Backpropagation, short for "backward propagation of errors," is a fundamental algorithm used to train artificial neural networks. It is a supervised learning method that enables the network to learn from its mistakes and improve its predictions by iteratively adjusting the network's internal parameters (weights and biases).

The process of backpropagation involves two main phases:

- **Feedforward Pass:**
  - Input data is fed into the neural network, starting from the input layer and moving forward through any hidden layers to the output layer.
  - Each neuron in a layer performs calculations, typically involving a weighted sum of its inputs and an activation function, to produce an output that is then passed to the next layer.
- **Backward Pass (Error Propagation and Weight Update):**
  - After the feedforward pass, the network's predicted output is compared to the actual target output, and the difference (error or loss) is calculated.
  - This error is then propagated backward through the network, from the output layer towards the input layer.
  - Using calculus, specifically the chain rule, the algorithm calculates the gradient of the loss function with respect to each weight and bias in the network. This gradient indicates how much each parameter contributes to the overall error.
  - Finally, the weights and biases are adjusted in the opposite direction of their respective gradients, using an optimization algorithm like gradient descent. This adjustment aims to minimize the loss function and reduce the error in future predictions.

This iterative process of feedforward and backward passes is repeated over many training examples (epochs) until the network's performance reaches an acceptable level of accuracy, meaning the error between predicted and actual outputs is minimized. Backpropagation is crucial for training complex deep learning models with multiple hidden layers, allowing them to learn intricate patterns and relationships within the data.

Word embeddings can be broadly categorized into **frequency-based and prediction-based methods**. Frequency-based methods, like TF-IDF and GloVe, analyze how often words appear together in a corpus. Prediction-based methods, such as Word2Vec, train neural networks to predict context words given a target word. [1, 2, 3]

Frequency-based methods:

- **TF-IDF (Term Frequency-Inverse Document Frequency):** This method assigns a weight to each word based on its frequency in a document and its rarity across the entire corpus. Words that are frequent in a specific document but rare in the overall corpus get higher weights.
- **GloVe (Global Vectors for Word Representation):** GloVe leverages global word-word co-occurrence statistics to create word embeddings. It aims to capture semantic relationships between words by analyzing how frequently they appear together in a large text corpus.

Prediction-based methods:

- **Word2Vec:** This technique uses a neural network to learn word embeddings by predicting the context of a word or predicting a word given its context. It has two main architectures:
    - **Skip-gram:** Predicts the surrounding words given a target word.
    - **CBOW (Continuous Bag-of-Words):** Predicts the target word based on its surrounding context words.
- **FastText:** Similar to Word2Vec, but it also considers subword information (n-grams of characters), which can help with handling out-of-vocabulary words.

- ## Other notable techniques:
- **BERT (Bidirectional Encoder Representations from Transformers):** While BERT is primarily known for its transformer-based architecture, it also generates contextualized word embeddings, meaning the embedding of a word can change depending on the context it appears in.
- **Doc2Vec:** An extension of Word2Vec that learns embeddings for entire documents or sentences. [