Practice Questions

Q1. In a standard Multi-Layer Perceptron (ANN), if an input layer has n nodes and the first hidden layer has m nodes, what is the total number of trainable parameters between these two layers, including biases?

Solution:
To connect every input node to every hidden node, we require $n \times m$ weights. Additionally, each neuron in the hidden layer requires exactly one bias term to perform the affine transformation $z = Wx + b$.

$$\text{Total Parameters} = (n \times m) + m = m(n + 1)$$

Correct Answer: $n \times m + m$

Q2. A Convolutional Neural Network (CNN) layer uses 32 filters of size 3×3 on an RGB input image (3 channels). How many trainable parameters are in this layer (assuming biases are used)?

Solution:
Each filter must match the depth (channels) of the input.

- Weights per filter: $3 \times 3 \times 3 = 27$
- Bias per filter: 1
- Total per filter: $27 + 1 = 28$

Total parameters for 32 filters: $32 \times 28 = 896$. **Correct Answer:** 896

Q3. Why does a Max-Pooling layer typically have zero trainable parameters?

Solution:
Pooling is a fixed mathematical operation used for spatial down-sampling. Unlike a convolutional layer, it does not use weights to extract features; it simply selects the maximum (or average) value within a sliding window. There are no values to be updated via backpropagation. **Correct Answer:** It uses a fixed mathematical operation with no weights to learn.

Q4. In a Multiple Linear Regression model with k independent variables (features), how many parameters are estimated during training?

Solution:
The model follows the equation $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$. Here, β_1 through β_k are the coefficients for the features, and β_0 is the intercept (bias).

$$\text{Total} = k \text{ coefficients} + 1 \text{ intercept} = k + 1$$

Correct Answer: $k + 1$

Q5. Calculate the number of parameters for a 1D Convolutional layer with 10 filters of size 5, acting on an input with 16 channels (with biases).

Solution:
The formula for a 1D conv layer is similar to 2D: (Kernel Size $\times$ Input Channels $+ 1$) $\times$ Number of Filters.

$$(5 \times 16 + 1) \times 10 = (80 + 1) \times 10 = 810$$

Correct Answer: 810

Q6. A Fully Connected (Dense) layer has 1024 input units and 512 output units. If we remove the bias terms, how many parameters are saved?

Solution:
In a Dense layer, the bias is applied to the output neurons. Since there are 512 output neurons, there are 512 bias terms. Removing them reduces the parameter count by exactly that amount. **Correct Answer:** 512

Q7. In a CNN, how does the "stride" parameter affect the total number of trainable parameters in a convolutional layer?

Solution:
Stride determines how the filter moves across the input, which affects the *output dimensions* (width and height), but it does not change the dimensions or the number of the filters themselves. **Correct Answer:** It has no effect on the number of trainable parameters.

Q8. Consider an ANN with an input of size 100, two hidden layers of 50 nodes each, and an output layer of 10 nodes. What is the total parameter count?

Solution:
We calculate layer by layer:

(a) Input to H1: $(100 \times 50) + 50 = 5050$

(b) H1 to H2: $(50 \times 50) + 50 = 2550$

(c) H2 to Output: $(50 \times 10) + 10 = 510$

Total: $5050 + 2550 + 510 = 8110$. **Correct Answer:** 8,110

Q9. Depthwise Separable Convolutions are used to reduce parameters. If a standard convolution has N filters of size $K \times K$ on M input channels, why are parameters reduced?

Solution:
Standard convolutions perform spatial filtering and channel combining simultaneously ($K \times K \times M \times N$). Depthwise separable convolutions split this into two steps: one filter per input channel (Depthwise) and then a 1×1 convolution to combine channels (Pointwise). This factorization breaks the multiplicative relationship between kernel size and the number of output filters. **Correct Answer:** They are reduced because we separate spatial filtering from channel combining.

Q10. A Softmax output layer has 10 nodes and receives input from a hidden layer with 128 nodes. How many parameters are in this specific output layer?

Solution:
Softmax is typically applied to the output of a Dense layer.

$$\text{Weights} = 128 \times 10 = 1280$$

$$\text{Biases} = 10$$

Total: $1280 + 10 = 1290$. **Correct Answer:** 1,290

Q11. In the Transformer architecture, what is the primary purpose of **Multi-Head Attention** compared to single-head attention?

- ☐ To reduce the number of trainable parameters in the model.
- ☐ To allow the model to jointly attend to information from different representation subspaces at different positions.
- ☐ To make the model recurrent so it can process sequences like an RNN.
- ☐ To act as a pooling layer to reduce dimensionality.

Solution: The second option is correct. Multi-head attention enables the model to focus on different types of linguistic relationships (e.g., coreference, syntax) simultaneously.

Q12. What does the **Attention Mask** specifically handle in a BERT input pipeline?

- ☐ It masks random words for the Masked Language Modeling (MLM) task.
- ☐ It prevents the model from attending to [PAD] tokens used to equalize batch lengths.
- ☐ It identifies the syntactic location of verbs and nouns.
- ☐ It encrypts the tokenized input for data privacy.

Solution: The second option is correct. The mask tells the self-attention mechanism to ignore the zero-padding used for batching.

Q13. In the context of NLP benchmarks, what is the **Spearman Rank Correlation** (ρ) actually measuring?

- ☐ The exact mathematical distance between two high-dimensional vectors.
- ☐ The monotonic relationship between the model's similarity rankings and human rankings.
- ☐ The inference latency of the model per token.
- ☐ The percentage of the vocabulary covered by the training set.

Solution: The second option is correct. It determines if the model ranks more similar pairs higher than less similar pairs, regardless of the absolute score.

Q14. What is the **"Vanishing Gradient"** problem in standard RNNs that LSTMs were designed to mitigate?

- ☐ Model weights overflow due to large learning rates.
- ☐ Gradients shrink exponentially as they are propagated back through time, causing the model to lose long-range dependencies.
- ☐ The model fails to recognize sub-word units.
- ☐ The Adam optimizer becomes unstable in deep architectures.

Solution: The second option is correct. LSTMs use gates to create a persistent "cell state" that bypasses the vanishing effect.

Q15. Why is **Positional Encoding** a mandatory component in Transformer models like BERT?

- ☐ Because Transformers process all tokens in parallel and have no inherent mechanism to model word order.
- ☐ To identify the source language of the input sequence.
- ☐ To normalize the magnitude of embeddings across different layers.
- ☐ To distinguish between training and validation data batches.

Solution: The first option is correct. Without this, "dog bites man" and "man bites dog" would yield identical representations.

Q16. In the **Word2Vec Skip-gram** architecture, what is the primary training objective?

- ☐ To predict a target word given its surrounding context.
- ☐ To predict the surrounding context words given a specific target word.
- ☐ To minimize the Euclidean distance between all words in a document.
- ☐ To classify the sentiment of a sentence based on its tokens.

Solution: The second option is correct. Skip-gram learns by using a single word to predict the words in its immediate window.

Q17. What problem does **Subword Tokenization** (e.g., WordPiece or BPE) solve in modern NLP?

- ☐ It reduces the total length of the input sequence.
- ☐ It solves the "Out-of-Vocabulary" (OOV) problem by decomposing unknown words into known meaningful sub-units.
- ☐ It removes frequent stop words like "the" and "is" automatically.
- ☐ It decreases the dropout rate in the hidden layers.

Solution: The second option is correct. This allows the model to process words like "un-happi-ness" even if it hasn't seen the specific full word during training.