Exercise sheet 4: Parameter-efficient Fine-tuning

Problem 1: Selecting PEFT configuration under parameter constraints [MG]

You are tasked with fine-tuning a small **encoder-only Transformer** using parameter-efficient methods, under a limited *trainable parameter* budget.

Model configuration The model to be fine-tuned has the following configuration:

- **Number of blocks:** $N_{\text{blocks}} = 2$
- **Number of attention heads per layer:** $h = 2$
- **Embedding dimension:** $d_{\text{model}} = 32$
- **Key/query/value dimension:** $d_k = d_q = d_v = 16$

PEFT methods You will use the following techniques:

1. **LoRA:** Applied to all query and key matrices, $\mathbf{W}^Q$ and $\mathbf{W}^K$, in every self-attention layer, with rank parameter r.
2. **Soft prompts:** Introduction of P soft prompt tokens.
3. **Adapters:** One adapter (without biases) is added after each attention layer. Each adapter is a feedforward network with one hidden layer of dimension d_a.
4. **IA^3:** Elementwise scaling inside attention (on keys and values):

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\left(\mathbf{l}_k \odot \mathbf{K}^\top\right)}{\sqrt{d_k}}\right)\left(\mathbf{l}_v \odot \mathbf{V}\right),$$

with a separate set of $\mathbf{l}_k$ and $\mathbf{l}_v$ vectors in each head of each attention layer.

Parameter limits You are given the following **limits** for the maximum number of trainable parameters you can have for these techniques:

- Maximum LoRA parameters: $N_{\text{LoRA}} = 1200$
- Maximum soft prompt parameters: $N_{\text{soft}} = 320$
- Maximum adapter parameters: $N_{\text{adapters}} = 1100$

Task 1: Compute the number of trainable parameters

Find the **largest even values** for

$$r, \quad P, \quad d_a$$

that meet the following constraints.

1. The chosen LoRa rank r ensures

$$\boxed{\text{total number of LoRA trainable parameters } \leq N_{\text{LoRA}}.}$$

2. The number of soft prompt tokens P ensures

$$\boxed{\text{total number of soft prompt trainable parameters } \leq N_{\text{soft}}.}$$

3. The hidden layer adapter dimension d_a ensures

$$\boxed{\text{total number of adapter parameters } \leq N_{\text{adapters}}.}$$

Also, compute the $\boxed{\text{total number of IA}^3 \text{ trainable parameters}}$ in the model.

Submission format: Submit a text file named `peft_config.txt` containing your chosen hyperparameters and computed parameter counts, formatted exactly as follows:

```
# LoRA (applied to W_Q and W_K in each block)
r = ...
num_trainable_lora = ...

# Soft Prompts
P = ...
num_trainable_soft = ...

# Adapters (one per attention layer)
d_a = ...
num_trainable_adapters = ...

# IA3 (keys and values in each block)
num_trainable_ia3 = ...
```

Exercise 2: Fine-tuning an LM [MG]

We work with the following indexed token vocabulary

$$\mathcal{V} = \{1 : \texttt{hi}, 2 : \texttt{hello}, 3 : \texttt{bye}, 4 : \texttt{regards}\}.$$

The associated token embeddings are as follows:

$$E(\texttt{hi}) = [1, 0], \quad E(\texttt{hello}) = [1, 0], \quad E(\texttt{bye}) = [0, 1], \quad E(\texttt{regards}) = [0, 1].$$

Consider a **simplified decoder-only LM** given by:

$$\begin{aligned}
\mathbf{Z} &= \text{softmax}(\mathbf{Q}\mathbf{K}^\top)\mathbf{V}\,\mathbf{W}^O && \text{(1) self-attention output (embeddings)}\\
\mathbf{H} &= \mathbf{X} + \mathbf{Z} && \text{(2) residual connection (embeddings)}\\
\mathbf{P} &= \text{Softmax}(\mathbf{h}_L\,\mathbf{W} + \mathbf{b}) && \text{(3) language modeling head (next-token probabilities)}
\end{aligned}$$

where:

- $\mathbf{X} \in \mathbb{R}^{L \times d_{\text{model}}}$: input embeddings.
- $\mathbf{W} \in \mathbb{R}^{d_{\text{model}} \times |\mathcal{V}|}$: weight matrix in the LM head.
- $\mathbf{b} \in \mathbb{R}^{1 \times |\mathcal{V}|}$: bias vector in the LM head.
- $\mathbf{h}_L \in \mathbb{R}^{1 \times d_{\text{model}}}$: representation of the last token in the sequence.

The attention weight matrices are: $\mathbf{W}_Q = \mathbf{W}_K = \mathbf{W}_V = \mathbf{W}^O = \mathbf{I}_2$.
The LM head weight matrix is (column order [$\texttt{hi}, \texttt{hello}, \texttt{bye}, \texttt{regards}$]):

$$\mathbf{W} = \begin{bmatrix} 4 & 3 & 0 & 0 \\ 0 & 0 & 4 & 3 \end{bmatrix}.$$

and the bias vector is a zero vector $\mathbf{b} = \{0\}^{|\mathcal{V}|}$.

Fine-tuning for behavior change Currently, the model responds to informal greetings such as `hi` with `hi`, and farewells such as `bye` with `bye`. More precisely, the model generates the following next-token probabilities:

$$P(\texttt{hi} \mid \texttt{hi}) \approx 1, \qquad P(\texttt{bye} \mid \texttt{bye}) \approx 1.$$

We wish to fine-tune the model so that it adopts a slightly more formal communication style, replying `hello` to `hi`, and `regards` to `bye`. Formally, the *fine-tuned model* should satisfy:

$$P_{\text{fine-tuned}}(\texttt{hello} \mid \texttt{hi}) \approx 1, \qquad P_{\text{fine-tuned}}(\texttt{regards} \mid \texttt{bye}) \approx 1.$$

To achieve this change in behavior without retraining the entire model, we will consider two **parameter-efficient fine-tuning (PEFT)** techniques:

1. **BitFit:** fine-tuning the bias vector $\mathbf{b} \in \mathbb{R}^{1 \times |\mathcal{V}|}$ only.
2. **LoRA:** low-rank adaptation of the LM head matrix $\mathbf{W} \in \mathbb{R}^{d_{\text{model}} \times |\mathcal{V}|}$ with rank $r = 1$.

Task 2: BitFit.

In BitFit, only the bias vectors get fine-tuned. Define a new bias vector $\mathbf{b}_{\text{fine-tuned}}$ that makes the model respond as intended:

$$P_{\text{fine-tuned}}(\texttt{hello} \mid \texttt{hi}) \approx 1, \qquad P_{\text{fine-tuned}}(\texttt{regards} \mid \texttt{bye}) \approx 1.$$

Task 3: LoRA (with rank $r = 1$) on W only.

With LoRA, fine-tuning the matrix $\mathbf{W}$ yields an *updated weight matrix*

$$\mathbf{W}' = \mathbf{W} + \Delta\mathbf{W}, \qquad \Delta\mathbf{W} = \mathbf{A}\mathbf{B},$$

where $\mathbf{A} \in \mathbb{R}^{d_{\text{model}} \times r}$, $\mathbf{B} \in \mathbb{R}^{r \times |\mathcal{V}|}$, and $r = 1$. The fine-tuned LM head computes next-token probabilities as follows:

$$\mathbf{P}_{\text{fine-tuned}} = \text{Softmax}\big(\mathbf{h}_L \mathbf{W}' + \mathbf{b}\big).$$

Choose matrices $\mathbf{A}$ and $\mathbf{B}$ which induce the following behavior:

$$P_{\text{fine-tuned}}(\texttt{hello} \mid \texttt{hi}) \approx 1, \qquad P_{\text{fine-tuned}}(\texttt{regards} \mid \texttt{bye}) \approx 1.$$

Report $\mathbf{A}$, $\mathbf{B}$ and the resulting $\mathbf{W}'$.

Submission format: Submit a single text file named `peft.txt` containing all parameter values, formatted exactly as follows (replace all placeholders with numbers):

```
# Task 2: BitFit (output bias)
b = [b_hi, b_hello, b_bye, b_regards]

# Task 3: LoRA (r=1) on W only
A = [[a1], [a2]]            # 2x1
B = [[b_hi, b_hello, b_bye, b_regards]]  # 1x4
Wprime = [[..., ..., ..., ...],
          [..., ..., ..., ...]]
```