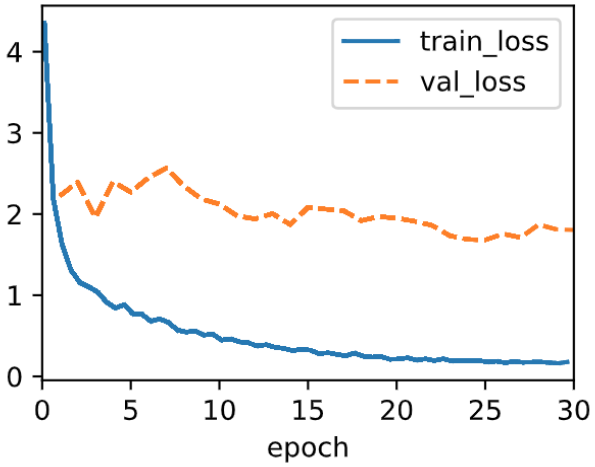
Jacinto Martinez | Student ID: 801135016

HW6

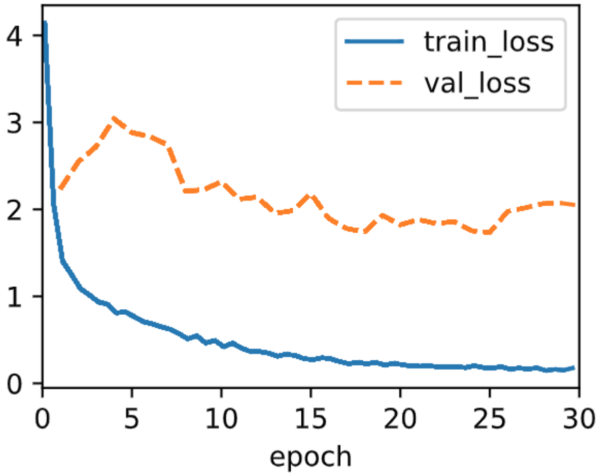
Github Link: https://github.com/jacintomart/4106/tree/main/HW6

Problem 1

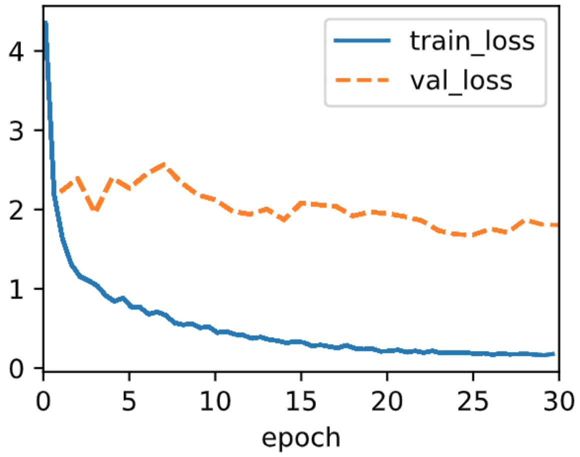
Baseline Transformer



3-Block Transformer



4-Block Transformer



The baseline translation transformer model was adjusted by increasing the number of encoder and decoder blocks. This adjustment resulted in a decrease in performance, with the validation loss of each model increasing with the number of blocks. The table below compares the loss and training time metrics of each of the 3 models trained, with a lower loss rank indicating lower loss.

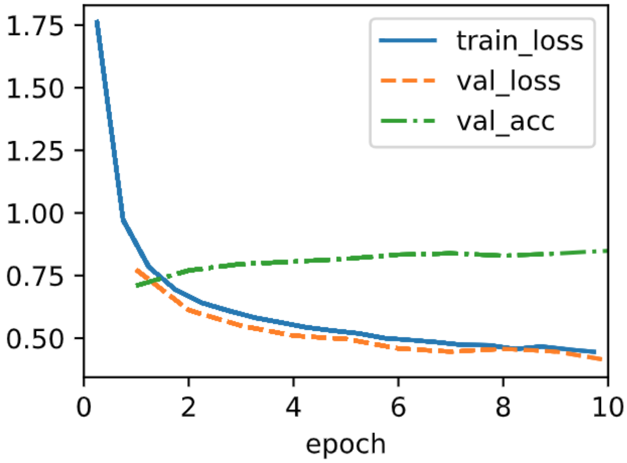
Model	**Loss Rank**	**Train Time**
2-Block Translator	1st	1 min 32 sec
3-Block Translator	2nd	2 min
4-Block Translator	3rd	2 min 13 sec

After training, each model was given 4 English sentences to translate to French. The translations from the models had similar BLEU (Bilingual Evaluation Understudy) scores, though the 4-block model did show a slight decrease in scores and all models showed similar complexity in translation quality. The table below shows the translations of each model along with their corresponding BLEU scores.

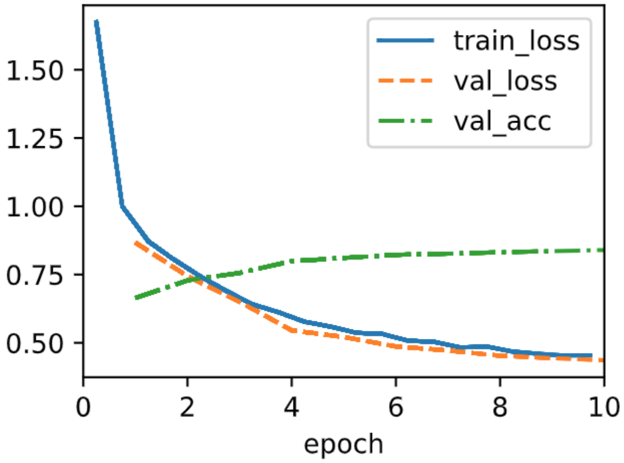
Model	**Sample English Sentence Translation**			
	‘go.’	**‘i lost’**	**‘he’s calm’**	**‘i’m home’**
2-Block	va! bleu: 1.000	j’ai perdu. bleu: 1.000	il est <unk>. bleu: 0.658	je suis chez moi bleu: 1.000
3-Block	va! bleu: 1.000	j’ai perdu. bleu: 1.000	il est mouillé. bleu: 0.658	je suis chez moi bleu: 1.000
4-Block	va,! bleu: 0.000	j’ai perdu. bleu: 1.000	il est mouillé. bleu: 0.658	je suis chez moi bleu: 1.000

Problem 2

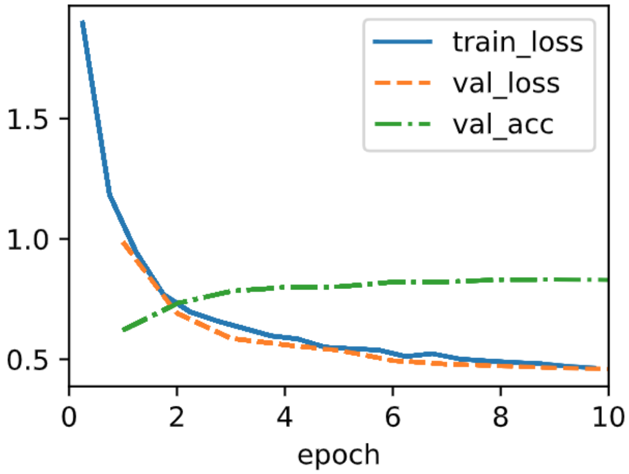
Baseline Vision Transformer



3-Block Vision Transformer



4-Block Vision Transformer



The number of hidden MLP nodes was reduced from the example provided in the textbook in order to achieve a feasible training time for baseline and adjusted models. Then, the baseline vision transformer model was adjusted by increasing the number of encoder and decoder blocks. These adjustments resulted in an overall similar performance across the 3 models trained, with only a slight decrease in validation accuracy. The table below compares the validation and training time metrics of each of the 3 models, with a lower validation rank indicating higher accuracy.

Model	Val Accuracy Rank	Train Time
2-Block Vision	1st	12 min 10 sec
3-Block Vision	2nd	17 min 5 sec
4-Block Vision	3rd	21 min 14 sec

Note: I was unable to run ptflops complexity measurements on the models trained due to errors with the way d2l assembled the models, as they were created entirely with d2l classes unlike previous homeworks where the model was first constructed using pytorch nn classes then passed to d2l initializers. Additional time would be needed in future iterations of the assignment to fix this issue and get accurate complexity metrics.