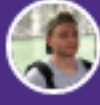
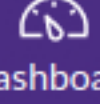


W


canvas




Account



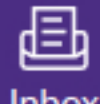
Dashboard




Courses




Calendar




Inbox



History



Help



Assignment 8: Embeddings, Recurrent Networks, and Sequences (Part 3)

New Attempt

Due

Jun 1 by 5:59pm

Points

6

Submitting

a text entry box or a file upload

Question 8 (1 point)

For this week's homework, we're using a Hugging Face Trainer wrapper to finetune and evaluate a PyTorch model (because pytorch supports gradient_accumulation_steps to accumulate gradients across training batches). [Perhaps someday Keras' model.fit() will also support a gradient_accumulation_steps argument.]

The console output for this week's Kaggle task includes print(model) output for the PyTorch model.

For my model, the console output can be found here: <https://www.cross-entropy.net/ML530/imdb-console.txt>

When I look at the "RobertaEmbeddings", I see the following entry for "word_embeddings":
(word_embeddings): Embedding(50265, 1024, padding_idx=1)

This means our vocabulary consists of 50,265 tokens, and each embedding consists of 1,024 features per token.

So the total number of parameters for "word_embeddings" is 51,471,360 (50,265 tokens * 1,024 features per token); and these parameters occupy 205,885,440 (51,471,360 parameters * 4 per bytes per float32 parameters) bytes of memory.

- For the model you choose ...
- a) Which model did you choose?
 - b) What is the size of the vocabulary?
 - c) How many features are used for each token?
 - d) How many parameters are there for the word embeddings?
 - e) How much memory is consumed to store these parameters? [Assume float32 representation for the parameters.]

Model 8 (5 points)

Please navigate to the following URL to accept the invitation for this Kaggle task:

<https://www.kaggle.com/t/4ff5f2bdd33e455a9ee9b1d85149850a>

Activate the pytorch conda environment on your VM
conda activate py37_pytorch

Install the Kaggle API and the Hugging Face transformers library into your pytorch environment:
pip install kaggle
pip install transformers

Download the data and the finetuning script:
kaggle competitions download ml530-2021-sp-imdb
wget --timeout=2 https://www.cross-entropy.net/ML530/imdb-train.py.txt

Pick one of the following models to build a model for sentiment classification, then use that name as an argument when running the imdb-train.py.txt script:

- bert-base-cased
- bert-base-uncased
- bert-large-cased
- bert-large-uncased
- roberta-base
- roberta-large
- distilbert-base-uncased
- distilbert-base-cased
- distilroberta-base
- microsoft/deberta-base
- microsoft/deberta-large
- squeezebert/squeezebert-uncased

You can find descriptions for these models here: https://huggingface.co/transformers/pretrained_models.html

The size of the "roberta-large" model is stated as "24-layer, 1024-hidden, 16-heads, 355M parameters". The "1024-hidden" means each embedding has 1024 features (so d_model = 1024). The larger the number of layers, the longer it will take to train. The large the number of features per embedding, the longer it will take to train. The "roberta-large" model takes around 6 hours for training and evaluation on a K80 GPU on a lab machine.

For example, to run roberta-large, use:
python imdb-train.py.txt roberta-large

Upload your predictions:
kaggle competitions submit ml530-2021-sp-imdb -f predictions.csv -m "imdb submission"

Submission

✓ Submitted!

Jun 7 at 8:53pm (late)

[Submission Details](#)

[Download HW08.txt](#)

Grade: 6 (6 pts possible)

Graded Anonymously: no

Comments:
No Comments