Wensong Hu
EECS
University of Michigan

March 21, 2024

HW4 — CNN, RNN, Transfer Learning and Transformers

1 CNN for Multi-class Classification

1.1 Implementation of `cnn_layers.py`

Submitted to Autograder.

1.2 Implementation of `cnn.py`

Submitted to Autograder.

1.3 Result

Plot see figure 1.

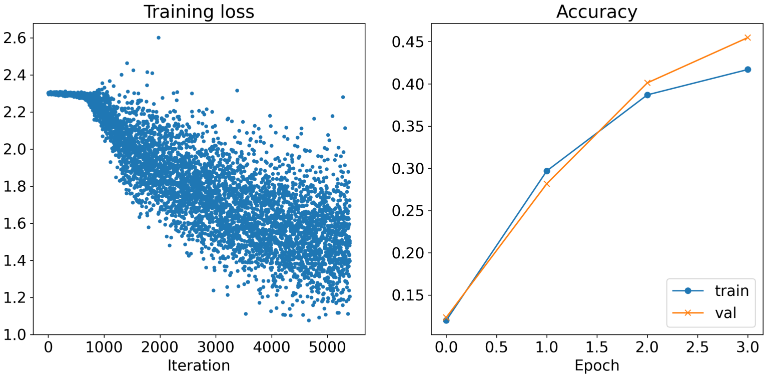


Figure 1: CNN loss and accuracy plot

Test accuracy is: 44.99%

2 Application to Image Captioning

2.1 Implementation of `rnn_layers.py`

Submitted to Autograder.

2.2 Implementation of `rnn.py`

Submitted to Autograder.

2.3 Result

Learning curves see figure 2.

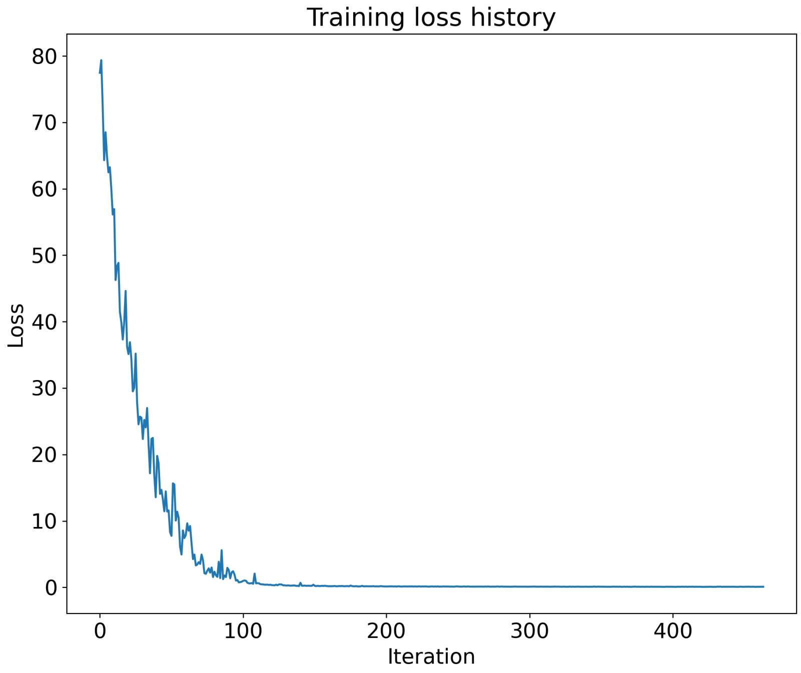
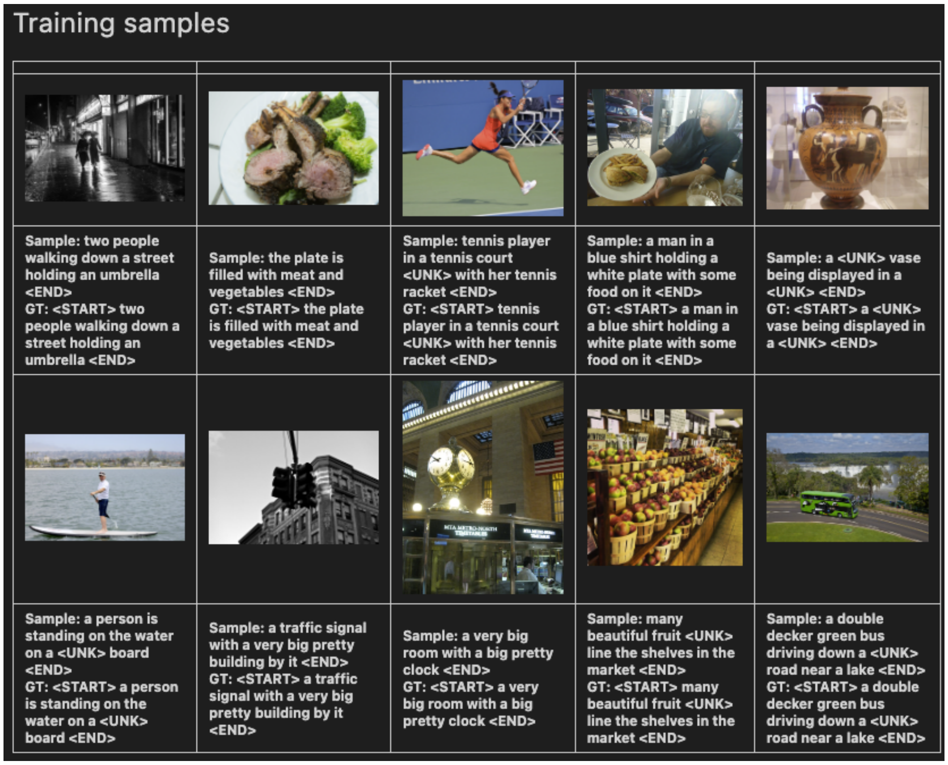


Figure 2: RNN loss plot

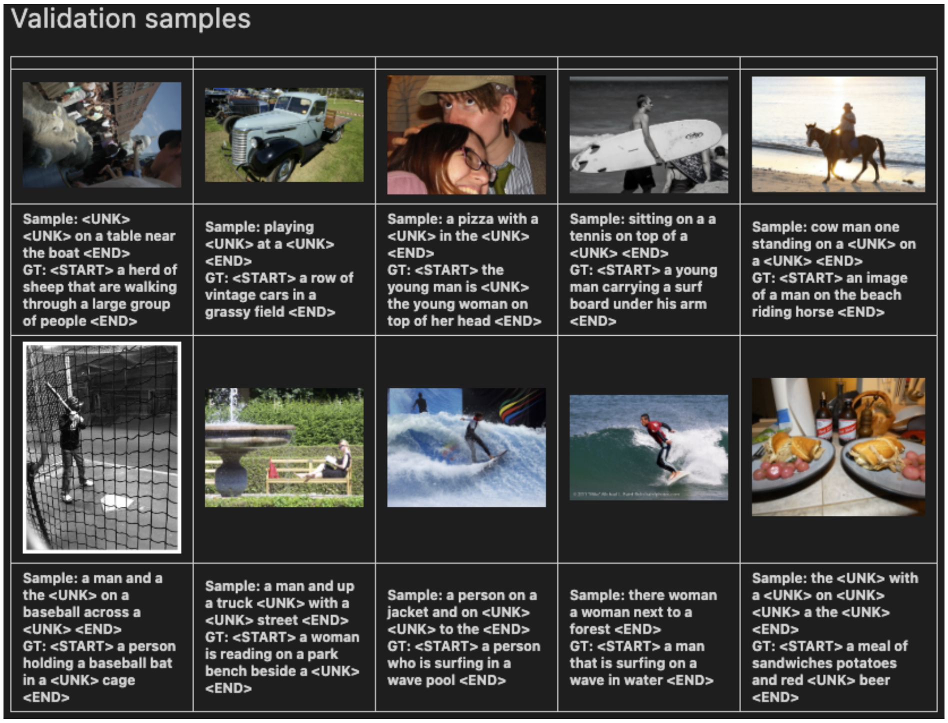
Caption test see figure 3 and 4



Training samples

Sample: two people walking down a street holding an umbrella <END> GT: <START> two people walking down a street holding an umbrella <END>	Sample: the plate is filled with meat and vegetables <END> GT: <START> the plate is filled with meat and vegetables <END>	Sample: tennis player in a tennis court <UNK> with her tennis racket <END> GT: <START> tennis player in a tennis court <UNK> with her tennis racket <END>	Sample: a man in a blue shirt holding a white plate with some food on it <END> GT: <START> a man in a blue shirt holding a white plate with some food on it <END>	Sample: a <UNK> vase being displayed in a <UNK> <END> GT: <START> a <UNK> vase being displayed in a <UNK> <END>
Sample: a person is standing on the water on a <UNK> board <END> GT: <START> a person is standing on the water on a <UNK> board <END>	Sample: a traffic signal with a very big pretty building by it <END> GT: <START> a traffic signal with a very big pretty building by it <END>	Sample: a very big room with a big pretty clock <END> GT: <START> a very big room with a big pretty clock <END>	Sample: many beautiful fruit <UNK> line the shelves in the market <END> GT: <START> many beautiful fruit <UNK> line the shelves in the market <END>	Sample: a double decker green bus driving down a <UNK> road near a lake <END> GT: <START> a double decker green bus driving down a <UNK> road near a lake <END>

Figure 3: RNN train samples



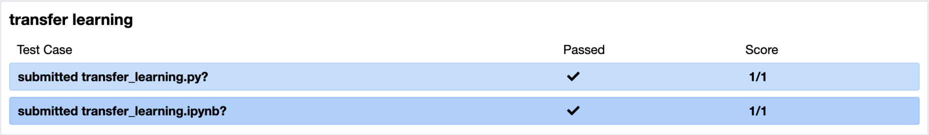
Validation samples

Sample: <UNK> <UNK> on a table near the boat <END> GT: <START> a herd of sheep that are walking through a large group of people <END>	Sample: playing <UNK> at a <UNK> <END> GT: <START> a row of vintage cars in a grassy field <END>	Sample: a pizza with a <UNK> in the <UNK> <END> GT: <START> the young man is <UNK> the young woman on top of her head <END>	Sample: sitting on a a tennis on top of a <UNK> <END> GT: <START> a young man carrying a surf board under his arm <END>	Sample: cow man one standing on a <UNK> on a <UNK> <END> GT: <START> an image of a man on the beach riding horse <END>
Sample: a man and a the <UNK> on a baseball across a <UNK> <END> GT: <START> a person holding a baseball bat in a <UNK> cage <END>	Sample: a man and up a truck <UNK> with a <UNK> street <END> GT: <START> a woman is reading on a park bench beside a <UNK> <END>	Sample: a person on a jacket and on <UNK> <UNK> to the <END> GT: <START> a person who is surfing in a wave pool <END>	Sample: there woman a woman next to a forest <END> GT: <START> a man that is surfing on a wave in water <END>	Sample: the <UNK> with a <UNK> on <UNK> <UNK> a the <UNK> <END> GT: <START> a meal of sandwiches potatoes and red <UNK> beer <END>

Figure 4: RNN validation samples

3 Transfer Learning

3.1 Implementation of transfer_learning.py



transfer learning

Test Case	Passed	Score
submitted transfer_learning.py?	✓	**1/1**
submitted transfer_learning.ipynb?	✓	**1/1**

Figure 5: Transfer learning submission

3.2 Train/val loss

Fine-tune train/val loss:

```
Performance of pre-trained model without finetuning
Training complete in 0m 0s
Best val Acc: 0.522876
Finetune the model
Epoch 0/4
----------
train Loss: 0.5971 Acc: 0.6885
val Loss: 0.3130 Acc: 0.8954
Epoch 1/4
----------
train Loss: 0.4838 Acc: 0.7623
val Loss: 0.4117 Acc: 0.8431
Epoch 2/4
----------
train Loss: 0.4566 Acc: 0.8033
val Loss: 0.2277 Acc: 0.9216
Epoch 3/4
----------
train Loss: 0.4719 Acc: 0.7664
val Loss: 0.4320 Acc: 0.8366
Epoch 4/4
----------
train Loss: 0.4498 Acc: 0.7910
val Loss: 0.4693 Acc: 0.8693
Training complete in 0m 9s
Best val Acc: 0.921569
```

Freeze and fine-tune train/val loss:

```
Performance of pre-trained model without finetuning
Training complete in 0m 0s
Best val Acc: 0.614379
Finetune the model
Epoch 0/9
----------
train Loss: 0.6775 Acc: 0.6107
val Loss: 0.6437 Acc: 0.6013
Epoch 1/9
```

```
----------
train Loss: 0.6422 Acc: 0.6311
val Loss: 0.5897 Acc: 0.6405
Epoch 2/9
----------
train Loss: 0.6032 Acc: 0.7049
val Loss: 0.5549 Acc: 0.6732
Epoch 3/9
----------
train Loss: 0.5834 Acc: 0.7459
val Loss: 0.4927 Acc: 0.7582
Epoch 4/9
----------
train Loss: 0.5514 Acc: 0.7336
val Loss: 0.4642 Acc: 0.8105
Epoch 5/9
----------
train Loss: 0.5126 Acc: 0.7787
val Loss: 0.4236 Acc: 0.8235
Epoch 6/9
----------
train Loss: 0.5134 Acc: 0.7787
val Loss: 0.4242 Acc: 0.8627
Epoch 7/9
----------
train Loss: 0.5426 Acc: 0.7295
val Loss: 0.4300 Acc: 0.8497
Epoch 8/9
----------
train Loss: 0.5043 Acc: 0.7869
val Loss: 0.4190 Acc: 0.8562
Epoch 9/9
----------
train Loss: 0.5094 Acc: 0.7787
val Loss: 0.4172 Acc: 0.8562
Training complete in 0m 11s
Best val Acc: 0.862745
```

3.3 Prediction result

Fine-tune prediction shown in 6:

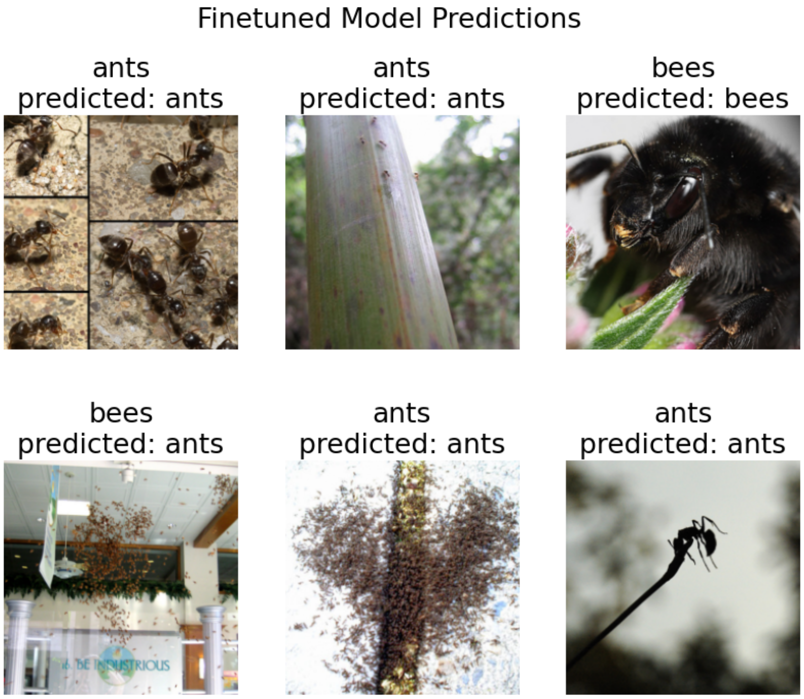


Figure 6: Fine-tuned model predictions

Freeze and fine-tune prediction shown in 7:

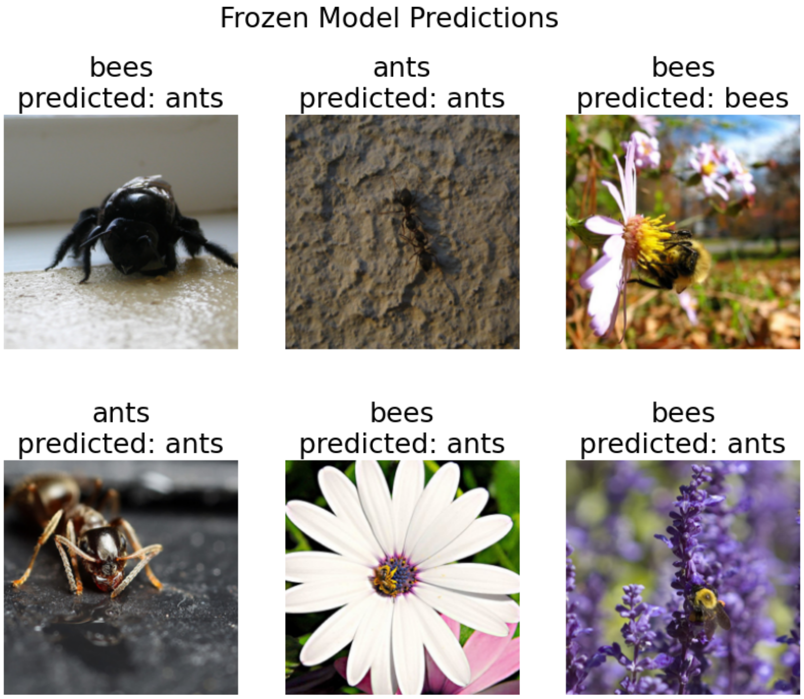


Figure 7: Freezed model predictions

4 Transformer Neural Networks

4.1 Implementation for `transformer.py`

Submitted to Autograder.

4.2 Math multiplication result

4.2.1 Final training and testing accuracy

Train accuracy: 99.86%
Test accuracy: 95.95%

4.2.2 Training loss plot

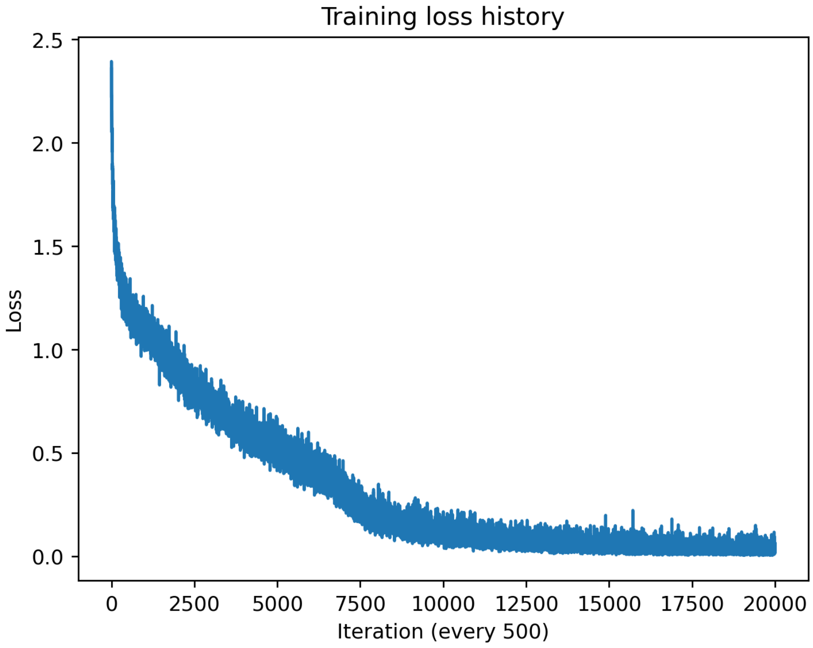


Figure 8: GPT math training loss plot

4.3 Tiny Stories

4.3.1 Final training loss

Final training loss: 0.6196

4.3.2 Three stroies

Story (1):
Once upon a time, there was a little girl named Emma. She loved to draw with her dictio-

nary and pretended to help the package outside from her dictionary. One day, Emma saw something very dictive. She wanted to stir the grass some with the pictures of her famous drewesting and smellled them all sorts. She wanted to play with her numbers, but her favorite was a bad dresss while she had a present. She asked, "Can I come to that pictures with you?" Her mom told her not to play with the toy anymore, and huggged her and said, "You can be the best tooy," Tom said, "You are welcome her mom." They huggged her back, thankful for the present. They huggged each other and clappped with the toy. The moral of the story is to always have fun and generous to be kind to it forever. The end.

Story (2):
Once there was an ordinary cat. The cat was very curious, but the talk had achieved. The cat wanted to come inside the house but not with the animals. The cat shouted "catch and became funny and warm." After a few minutes of colorful cats, the boy was very excited and felt satisfied. He saw that the cat was a big house with a hipppo. He asked his mom if he could hurt his kind friend for helping him buy something else. Finally, his mom said, "Yield, let's clean the cat!" They both clappped and clappped and sat back in the garden. They ate and laughed until they could hear the noise. They were happpy to help the new friend.

Story (3):
One day, the farmer convered her book with her, for trouble. She looked out the window and saw a beautiful book. The book was so pretty that it was cool, and it was her dream. The book said, "Thank you, Tim. I will try to make her sausages for your book." The mouse was proud that the boook liked her boook. The boook held her up the boook, was swinging towards its nice! Tim and Sam became best friends.

Submitted by Wensong Hu on March 21, 2024.