COMP 543 Assignment #6: Deep Learning with TensorFlow

Description

The learning objectives of this assignment are to give you experience specifying neural network architectures, training the neural networks, and seeing how well they perform on an image recognition task.

In this assignment, you will be using Google's open-source TensorFlow machine learning tool via Keras to implement some deep learning architectures that will be used to classify images as one of four different cell types.

Read the entire assignment before starting!

Honor Code

You make work fully collaboratively with your team members. You may discuss the assignment at a high level with other people in the course. However, you may not share code in any way (viewing, verbally, etc.) with non-team members. You may use the Internet to look up syntax for Keras commands but may not use it to search for answers to the problems. If you are unsure of what is allowed, ask!

Working with a Partner

For this assignment, unlike previous assignments, you are welcome to work with a partner. You don't have to work with a partner; the assignment is doable without one. We estimate 10-15 hours of work for an average student to complete all of the tasks. If you do work with a partner, make sure to put **both** partner's names in the write-up you submit. And only **one** partner should turn in the assignment.

AWS (review of TensorFlow Lab)

This assignment must be completed on AWS. Please do not use comp543.clear.rice.edu as the machine does not have sufficient power.

Running TensorFlow on AWS

Follow these steps:

1. Log on to Amazon AWS Educate and go to your AWS console
2. Click on "EC2"
3. Click on "Launch Instance". You will be asked for a machine instance to start up (this will govern the software on the machine you run).
4. Scroll down to "Deep Learning AMI (Ubuntu)" and click "Select". This machine instance (not surprisingly, given the name) has a number of deep learning tools installed on it.
5. Choose the machine type you will rent. Choose "t2.xlarge"
6. Click "Review and Launch"
7. Click "Launch"
8. Select the appropriate key file and acknowledge access

9. You can find your running instance by going to the EC2 Dashboard and then clicking on “Running Instances”.

10. Copy the images to the machine using `scp`. The `-r` flag will copy recursively

11. SSH into your machine (just like for EMR), You can find the connection string to connect to your machine by clicking on the “Connect” button

12. Start tensorflow:

    ```
    source activate tensorflow_p36
    ```

13. Run your code interactively or in batch using python

14. As usual, when you are done with your machine, **MAKE SURE TO SHUT IT DOWN!!**

The learning problems we’ll consider will take about 30 minutes each using one of these machines

What to Turn in

Submit

1. A python file (.py) with your code

2. A filled-out Excel spreadsheet with your write-up.

3. A text capture of the output of your running code output. You may submit a separate file for each model, if you like.

The Data

There are 12,515 images, each 320 x 240 pixels in size. Each image contains one cell type, either eosinophil, lymphocyte, monocyte, or neutrophil. The image labels are contained in bloodcellLabels.csv.

The dataset may be found here:

`https://www.kaggle.com/paultimothymooney/blood-cells`

Download the data. Use the dataset2-master files. Train your model on the images in the folder named `TRAIN` and test your model on the images in the folder named `TEST`. We will not be using the images in the folder named `TEST_SIMPLE`.

Each image contains an RGB (red-green-blue) channel value for each pixel.

Our Results

For comparison, the results we obtained on this assignment were:

Type	time per epoch (sec)	loss	training accuracy	testing accuracy
Dense FFNN	13.23	1.3863	0.2494	0.2505
CNN	15.46	0.0013	1.0000	0.4962
Modified CNN	19.37	2×10^{-5}	1.0000	0.4757
Our lazy best CNN	17.77	0.0351	0.9879	0.7692

The Code

We have provided you with code that reads the images in by RGB channel as well as skeleton code to run your models.

You will need to construct deep learning networks that generate sets of probabilities for each image, indicating how likely each image is to contain each of the four cell types.

Output

Once you have your code running, you can run it directly from the command line and redirect the output to both a file and the screen using the following command (Note: there is no space between > and &:

```
python myprogramname.py >&1 | tee myoutputfilename.txt
```

The Tasks

There are 4 separate tasks that you need to complete to finish the assignment.

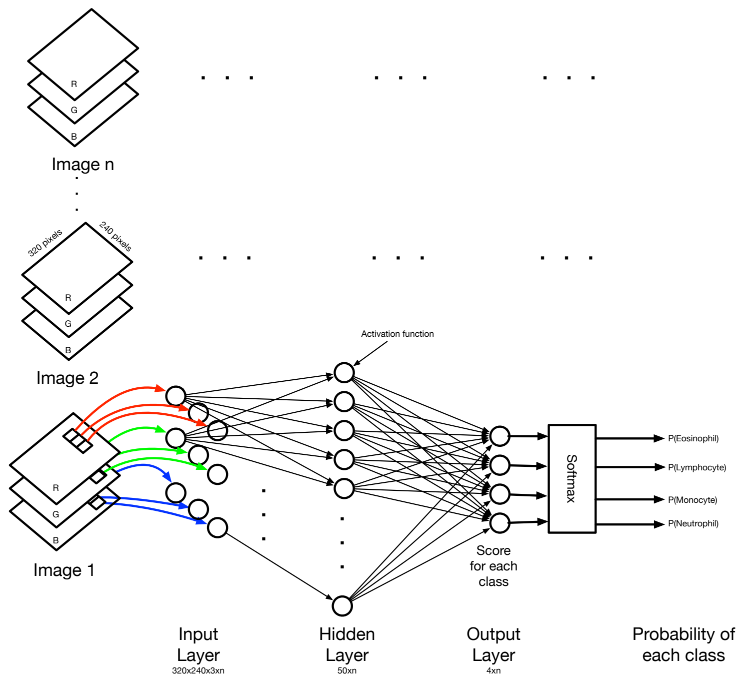
1 Task 1: Feed forward network

1.1 A: (20 points) Construct a feed forward network

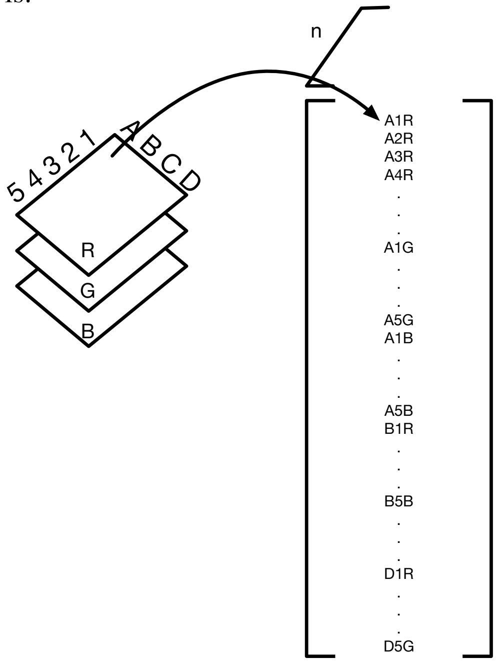
Construct a single hidden layer feed forward network with the following parameters / characteristics:

- Downsample the images via average pooling by a factor of 4, using ‘same’ padding
- 16 nodes in the hidden layer
- relu activation function
- 100 training epochs
- Batch size of 64
- Adam optimizer
- Compute accuracy as an additional metric
- Categorical_crossentropy loss function
- Softmax output layer

Train your model.
Be sure to time the training step, as you will need to report this information.
This figure depicts the components of a basic feed forward network.



Note that you need a mapping from the image pixels to the input layer. Many such mappings are possible. One is:



which can be obtained with the `Flatten` function in Keras.

1.2 B: (10 points) Compute results for your feed forward network

Please record your answers in the provided spreadsheet. Questions are repeated here for your reference. You may compute metrics (e.g. sensitivity, specificity, etc.) using equations in the spreadsheet, or you may write code to compute the metrics.

Please do NOT modify the results spreadsheet (other than possibly adding formulas) and please use a single cell for each answer. This format should help us combine results for reporting.
For this network:

1. What was the accuracy on the training data at the final epoch?
2. Time the training step. How many minutes did it take to train your model?
3. Populate the following confusion matrix, showing the counts for each class

		Predicted				
		Eosinophil	**Lymphocyte**	**Monocyte**	**Neutrophil**	**Total**
Actual	**Eosinophil**					
	Lymphocyte					
	Monocyte					
	Neutrophil					
	Total					

4. Fill in the following table of metrics

Cell Type	**% Correct**	**Sensitivity (Recall)**	**Specificity**	**Precision**	**F1**	**F2**
Eosinophil						
Lymphocyte						
Monocyte						
Neutrophil						

5. Which cell type was most predictable? Justify your answer.
6. If you worked with a partner, describe the breakdown of work:

2 Task 2: Convolutional Neural Network

2.1 A: (10 points) Create the CNN

Starting with the feed forward network you created, replace the dense layer with a convolution layer with a 3 x 3 kernel and `same` padding.

Train your model.

2.2 B: (10 points) Compute results for your CNN

Please record your answers in the provided spreadsheet. Questions are repeated here for your reference. You may compute metrics (e.g. sensitivity, specificity, etc.) using equations in the spreadsheet, or you may write code to compute the metrics.

1. What was the accuracy on the training data at the final epoch?
2. Time the training step. How many minutes did it take to train your model?
3. Populate the following confusion matrix, showing the counts for each class

		Predicted				
		Eosinophil	**Lymphocyte**	**Monocyte**	**Neutrophil**	**Total**
Actual	**Eosinophil**					
	Lymphocyte					
	Monocyte					
	Neutrophil					
	Total					

4. Fill in the following table of metrics

Cell Type	**% Correct**	**Sensitivity (Recall)**	**Specificity**	**Precision**	**F1**	**F2**
Eosinophil						
Lymphocyte						
Monocyte						
Neutrophil						

5. Which cell type was most predictable? Justify your answer.
6. If you worked with a partner, describe the breakdown of work:
7. Which network performed better, the feed forward or the convolutional? Justify your answer.

3 Task 3: Modify the CNN

3.1 A: (10 points) Modify the CNN

There are a number of ways in which the existing network can be improved. Make a **single** change to the network and recompute the confusion matrix and metrics. Possible changes include:

- Change the size of the images being used (specify)
- Change the size of the convolution kernel (specify)
- Handle image edges differently (specify)
- Change the number of hidden nodes (specify)
- Change the activation function (specify)
- Change the optimizer (specify)
- Add another convolutional layer (specify where, how many nodes, activation function used, kernel size, padding)
- Add a pooling layer
- Add a dropout layer (specify)
- Add a batch normalization layer
- Change the learning rate (specify)

Check the course discussion to ensure that your (team's) change is unique. Multiple teams may make similar changes (e.g. the number of hidden nodes) but each team making the same type of change must use a different parameter value. The first team to claim a modification and parameter value will get credit for that change.

3.2 B: (10 points) Compute results for your modified CNN

1. Describe the single change you made.
2. Why did you choose this change? For example, if you change the optimizer, why do you think the new optimizer will be better?
3. What made you believe it would improve the model?
4. In what way(s) did you expect it to impact the model?
5. What was the accuracy on the training data at the final epoch?
6. Time the training step. How many minutes did it take to train your model?
7. Populate the following confusion matrix, showing the counts for each class

		Predicted				
		Eosinophil	**Lymphocyte**	**Monocyte**	**Neutrophil**	**Total**
Actual	**Eosinophil**					
	Lymphocyte					
	Monocyte					
	Neutrophil					
	Total					

8. Fill in the following table of metrics

Cell Type	**% Correct**	**Sensitivity (Recall)**	**Specificity**	**Precision**	**F1**	**F2**
Eosinophil						
Lymphocyte						
Monocyte						
Neutrophil						

9. Which cell type was most predictable? Justify your answer.
10. If you worked with a partner, describe the breakdown of work:
11. Did this model perform better than the basic CNN? Justify your answer.

4 Task 4: Improve your CNN

Make **at least** one more change to your modified CNN to try and improve the model. If you feel that the results for your modified CNN decreased performance, you may start over with the basic CNN and make at least two changes.

The goal here is to see how good you can get your model to perform.

You can make a number of single changes, progressively (hopefully) improving the model, or you may make multiple changes at once. When you make multiple changes at the same time it can be difficult to tell which change caused the new results.

4.1 A: (10 points) Improve the CNN

There are a number of ways in which the existing network can be improved. Make **changes** to the network and recompute the confusion matrix and metrics.

4.2 B: (10 points) Compute results for your improved CNN

Give us your best results below. You do not need to report results for intermediary models (though you may find it beneficial to record them. If you record intermediary results, please do so on a second spreadsheet and submit it with your assignment. It will be interesting to see what you did.

1. Describe the change(s) you made.
2. Why did you choose these changes? For example, if you change the optimizer, why do you think the new optimizer will be better?
3. What made you believe they would improve the model?
4. In what way(s) did you expect it to impact the model?
5. What was the accuracy on the training data at the final epoch?
6. Time the training step. How many minutes did it take to train your model?
7. Populate the following confusion matrix, showing the counts for each class

		Predicted				
		Eosinophil	**Lymphocyte**	**Monocyte**	**Neutrophil**	**Total**
Actual	**Eosinophil**					
	Lymphocyte					
	Monocyte					
	Neutrophil					
	Total					

8. Fill in the following table of metrics

Cell Type	**% Correct**	**Sensitivity (Recall)**	**Specificity**	**Precision**	**F1**	**F2**
Eosinophil						
Lymphocyte						
Monocyte						
Neutrophil						

9. Which cell type was most predictable? Justify your answer.
10. If you worked with a partner, describe the breakdown of work:
11. Did this model perform better than your modified CNN? Justify your answer.