4-6 Wednesday – 210-GD3

Special topics in Computer Science INT3121 20

Lecturer: Nguyen Thi Ngoc Diep, Ph.D.

Email: ngocdiep@vnu.edu.vn

Slide & Code: https://github.com/chupibk/INT3121-20

Image classification with convolutional neural networks

Week	Content	Class hour	Self-study hour
1 28/8/2019	Introduction Image classification problem and its applications A toy problem with CIFAR10	2	1
2 (4/9/2019)	CNN model architectures and visualization	2	1
3 (11/9/2019)	Training and tuning parameters Automatic parameter learning	2	1
4 (18/9/2019)	Data augmentation Data generator	2	2-6
5	Transfer learning	2	2-6
6	Multi-output image classification	2	2-6
7	Building a training dataset How to write a report	1	2-6
8, 9, 10, 11	Seminar: Bag of tricks with CNN (as mid-term tests)	1	2-6
12, 13, 14	Final project presentations	1-3	2-6
15	Class summarization	1	open

Week 3 recall

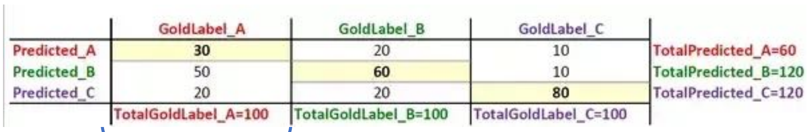
- Steps in training a CNN model:
  - Prepare data, x:(b,h,w,c), y(b,n)
  - Define layers
  - Compile the model
    - Loss: cross entropy loss
    - Optimizer: ADAM, RMSProp, GD, AdaGrad, SGD, Momentum
    - Evaluation metrics: accuracy
  - Train the model
    - Feed data and do weight updating
- Hyperparameter search
  - Hyperas

Addition to week 3: Evaluation metrics for multi-class classification

	Predicted YES	Predicted NO
Actual: YES	True positive	False Negative
Actual: NO	False Positive	True negative

Recall = TP/Total actual positive

Precision = TP/Total predicted positive



	GoldLabel_A	GoldLabel_B	GoldLabel_C	
Predicted_A	30	20	10	TotalPredicted_A=60
Predicted_B	50	60	10	TotalPredicted_B=120
Predicted_C	20	20	80	TotalPredicted_C=120
	TotalGoldLabel_A=100	TotalGoldLabel_B=100	TotalGoldLabel_C=100	

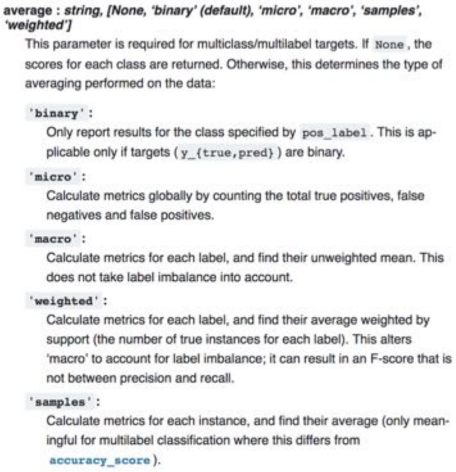
Recall label X = TP_X / Total actual X

Precision label X = TP_X / Total predicted_X

Ref: http://text-analytics101.rxnlp.com/2014/10/computing-precision-and-recall-for.html

sklearn.metrics

sklearn.metrics.precision_score(*y_true*, *y_pred*, *labels=None*, *pos_label=1*, *average='binary'*, *sample_weight=None*)



average : *string, [None, 'binary' (default), 'micro', 'macro', 'samples', 'weighted']*

This parameter is required for multiclass/multilabel targets. If `None`, the scores for each class are returned. Otherwise, this determines the type of averaging performed on the data:

`'binary'`:
Only report results for the class specified by `pos_label`. This is applicable only if targets (`y_{true,pred}`) are binary.

`'micro'`:
Calculate metrics globally by counting the total true positives, false negatives and false positives.

`'macro'`:
Calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.

`'weighted'`:
Calculate metrics for each label, and find their average weighted by support (the number of true instances for each label). This alters 'macro' to account for label imbalance; it can result in an F-score that is not between precision and recall.

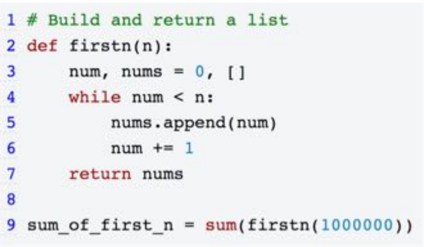
`'samples'`:
Calculate metrics for each instance, and find their average (only meaningful for multilabel classification where this differs from `accuracy_score`).

Data generator

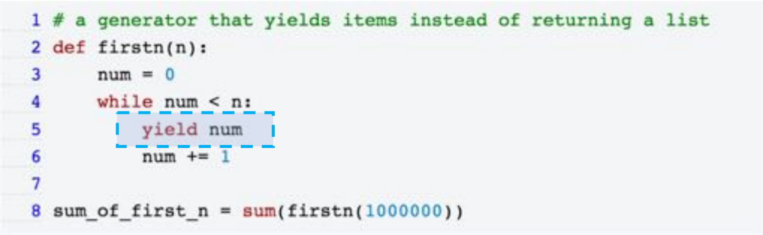
Python generator

- A generator is a function that behaves like an iterator
- But:
  - Does not hold results in memory
    - Values are generated when used
    - Lazy evaluation
  - Only iterate over them once
  - May take longer to run

Building a list

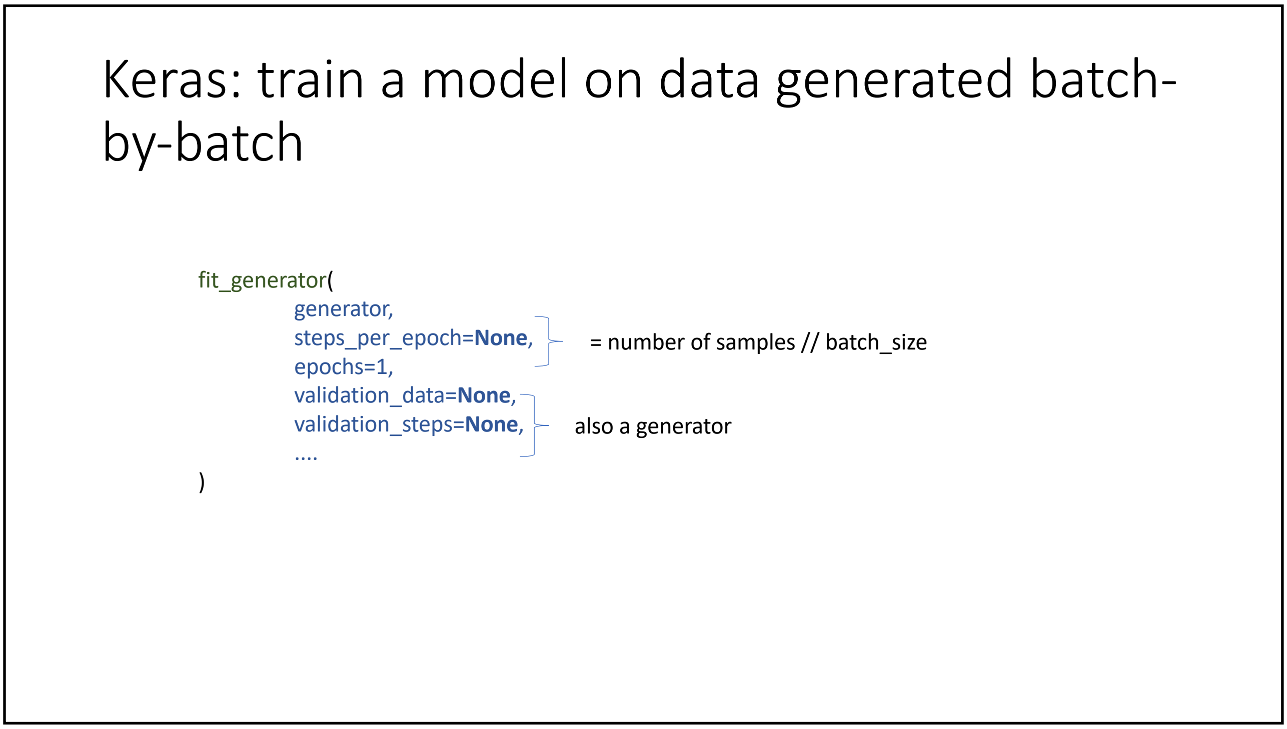


```python
# Build and return a list
def firstn(n):
    num, nums = 0, []
    while num < n:
        nums.append(num)
        num += 1
    return nums

sum_of_first_n = sum(firstn(1000000))
```



```python
# a generator that yields items instead of returning a list
def firstn(n):
    num = 0
    while num < n:
        yield num
        num += 1

sum_of_first_n = sum(firstn(1000000))
```

Ref: https://wiki.python.org/moin/Generators



Keras: train a model on data generated batch-by-batch

```
fit_generator(
        generator,
        steps_per_epoch=None,        = number of samples // batch_size
        epochs=1,
        validation_data=None,
        validation_steps=None,      also a generator
        ....
)
```

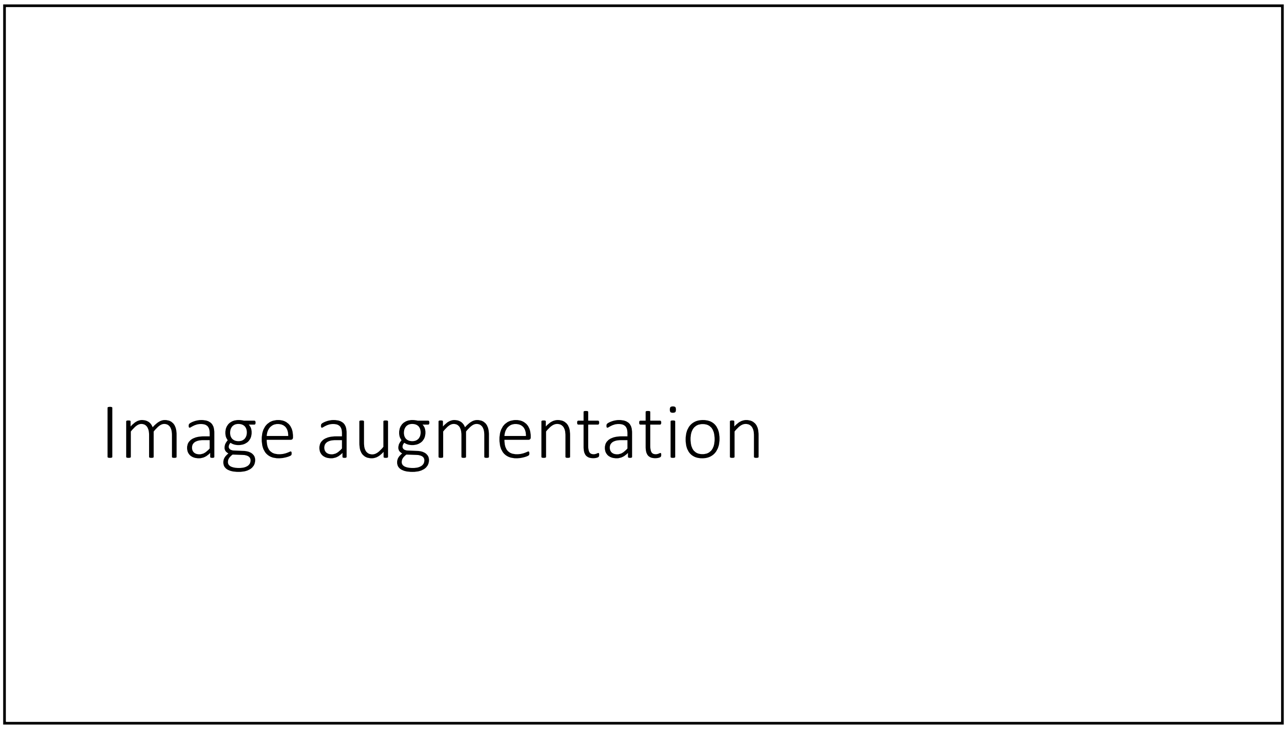
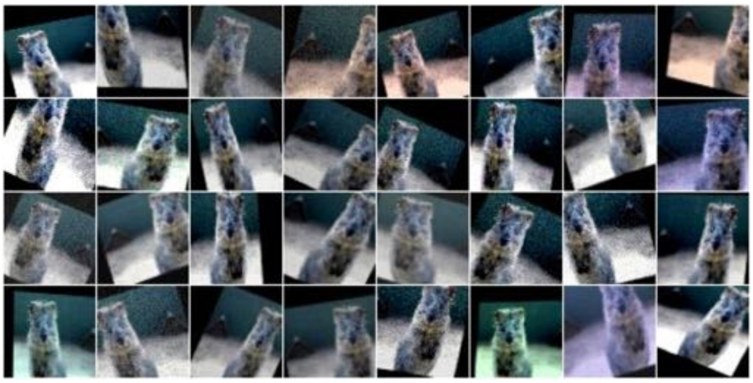


Image augmentation

Why augmenting images?



Ref: https://imgaug.readthedocs.io/en/latest/source/examples_basics.html

Types of augmentation

- Blur
- Sharpen
- Edge detection
- Adding noise
- Combining images
- Invert
- Contrast normalization
- Affine transformation
- Pad
- Crop
- Flip
- Resize
- Change color space
- ...

imgaug:
https://imgaug.readthedocs.io/en/latest/index.html

keras.preprocessing.image.ImageDataGenerator
https://keras.io/preprocessing/image

Augmentation + Generator

- Using keras: (https://keras.io/preprocessing/image/)
  - Using .flow(x,y)
  - or .flow_from_directory(directory)
- Self build:
  - Build a generator for data
  - For each batch of data, run the augmentation

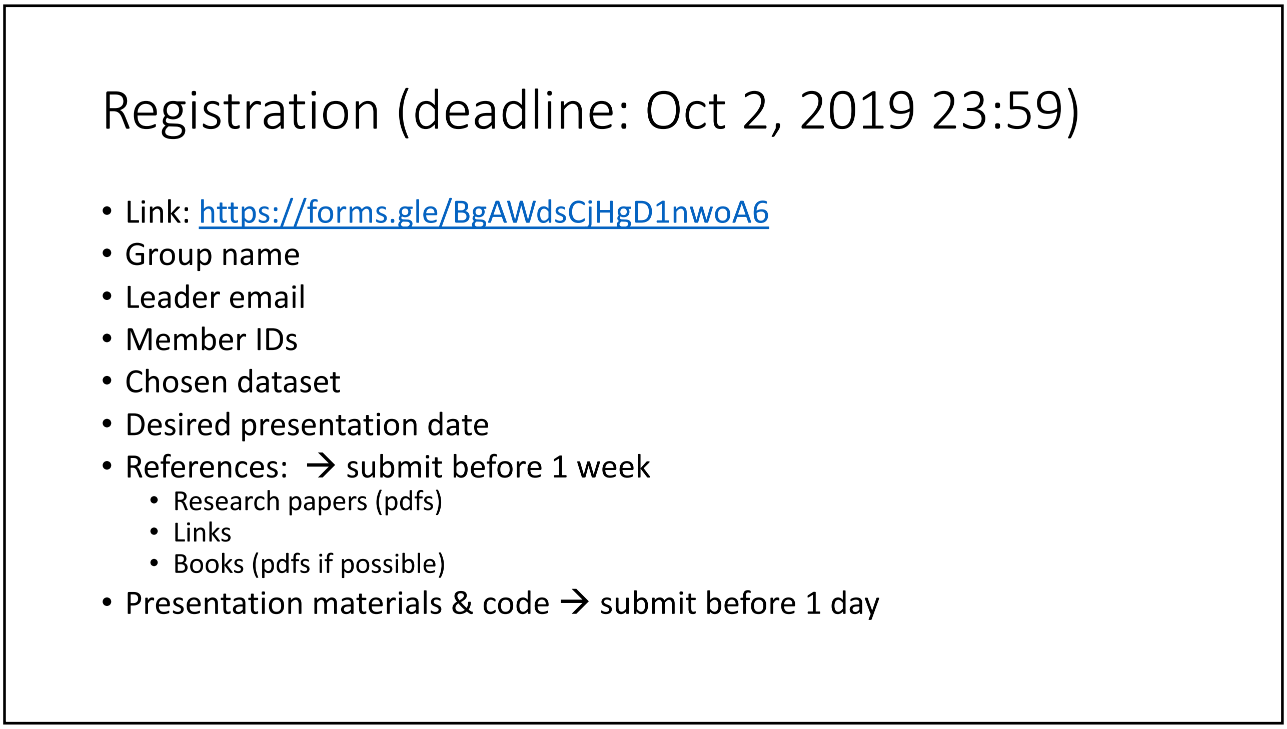
Towards mid-term projects

Registration for mid-term project

- Group: 3-5 members
- Work:
  - Choose a dataset
  - Survey of techniques have been used in order to improve the performance
  - Implement a classification model that applies those techniques
    - Reimplementation is okay
    - Combining is better
  - Analyze the results

Marking policy

- Comparisons of methods/techniques → 30%
  - More is better
- Implementation → 20%
- Analysis of the results → 20%
- Presentation → 10%
- Audiences’ comments → 20%



Registration (deadline: Oct 2, 2019 23:59)

- Link: https://forms.gle/BgAWdsCjHgD1nwoA6
- Group name
- Leader email
- Member IDs
- Chosen dataset
- Desired presentation date
- References: → submit before 1 week
  - Research papers (pdfs)
  - Links
  - Books (pdfs if possible)
- Presentation materials & code → submit before 1 day