★ Assignment 1

⇒ Goals:

* Understand the basic Image Classification pipeline & data-driven approach (train/predict stages)
* Understand the train/Val/test splits.
  ↳ Use of validation data for hyperparameter tuning.
* Develop proficiency in writing efficient vectorized code with numpy.
* Implement and apply a K-Nearest Neighbor (KNN) classifier
* Implement and apply a multiclass SVM classifier.
* Implement & apply Softmax classifier.
* Implement & apply Two layer neural network classifier
* Understand the difference & tradeoffs between these classifiers.
* Get a basic understanding of performance improvements from using higher-level representation than raw pixels.

{Color Histogram, Histogram of Gradients (HOG) features}

★ Setup Instructions

{Working remotely on Google Colaboratory}

{Combination of Jupyter notebook and Google Drive}

⇒ GC runs entirely in the cloud & comes preinstalled with many packages (e.g PyTorch & Tensorflow), so everyone has access to the same dependencies.

⇒ Colab benefits from free access to hardware accelerators like GPUs (K80, P100) and TPUs.

1. KNN Classifier (knn.ipynb)

★ CIFAR-10 dataset

10 → 10 classes

CIFAR → Canadian Institute for Advanced Research

⇒ Contains 60,000 32×32×3 color images in 10 class.

→ 6,000 image per class

→ (50,000 training image) + (10,000 test image)

⇒ The dataset is divided as follows:

~~① training batches~~

① 1 - test batch

→ 1000 randomly selected images for each class.

② 5 - training batches

→ 5 sets of 10,000 randomly selected images from the remaining images.

⇒ Class :

1. aeroplane	6. dog
2. automobile	7. frog
3. bird	8. horse
4. cat	9. ship
5. deer	10. truck

{The classes are completely mutually exclusive, no overlap}

⇒ Each file in dataset is a Python "pickled" object produced with cPickle.

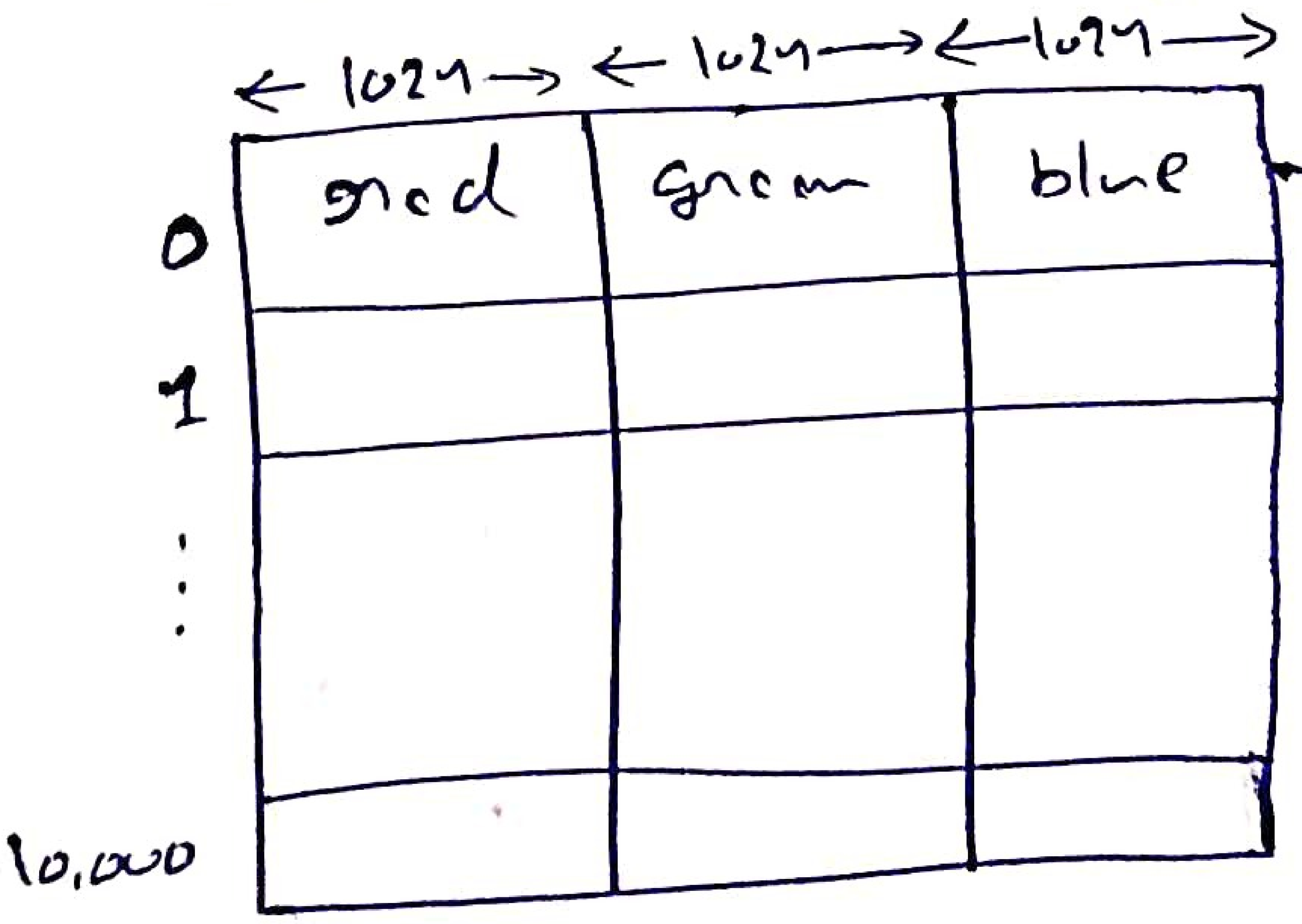
⇒ Python 3 routine which will open such file & return a dictionary:

```
def unpickle(file):
    import pickle
    with open(file,'rb') as fo:
        dict = pickle.load(fo, encoding='bytes')
    return dict
```

⇒ Each of the batch files contain a dictionary with the following elements:

- data

  ↳ 10,000 × 3072 numpy array of unit8s.



One row contains one image, stored in row-major order.

- labels

  ↳ A list of 10,000 numbers in the range 0-9.

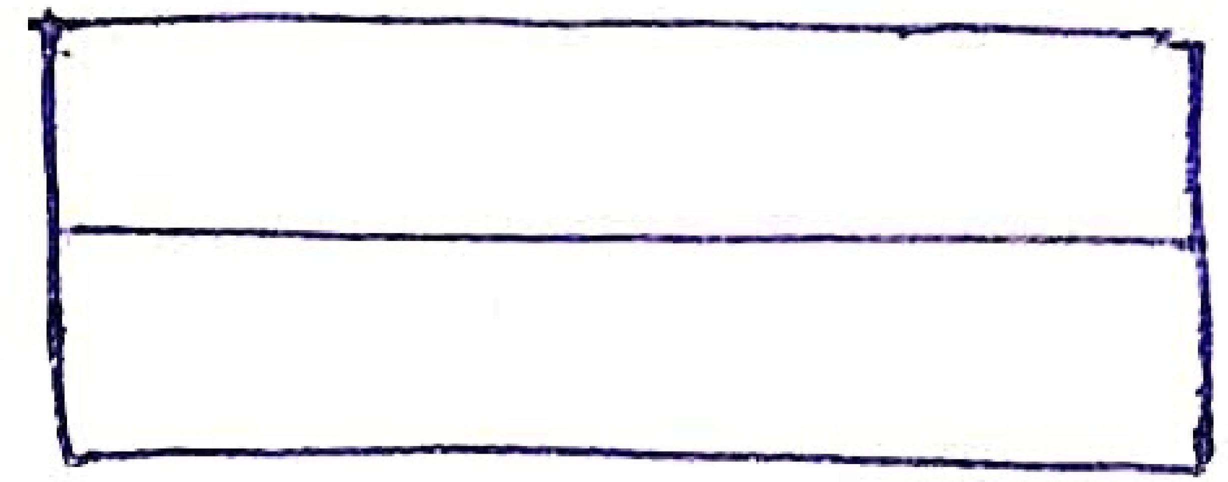
⇒ batches.meta, contain the following ~~dictionary~~ dictionary:

- label_name

  ↳ 10-element list which gives meaningful names to the numeric labels in the labels array.

Inline Question 1

* black → small distance
* white → large distance



(a) distinctly bright row

→ The test image corresponding to that row is far away from every train image.

(b) distinctly bright column

→ The train image corresponding to that column is far away from every test image.

Inline Question 2

- $P_{ij}^{(K)}$ → Pixel value at location (i, j) of some image I_K

h → height of images

w → width of images

n → Total number of images

$$\mu = \frac{1}{nhw} \sum_{K=1}^{n} \sum_{i=1}^{h} \sum_{j=1}^{w} P_{ij}^{(K)}$$ { Mean across all Pixels over all images }

$$\mu_{ij} = \frac{1}{n} \sum_{K=1}^{n} P_{ij}^{(K)}$$ { Pixel-wise mean μ_{ij} across all images }

⇒ σ & σ_{ij} are defined similarly

(1) $$\tilde{P}_{i,j}^{(K)} = P_{i,j}^{(K)} - \mu$$

$$dist_{i,j} = \tilde{P}_{i,j}^{(K_2)} - \tilde{P}_{i,j}^{(K_1)}$$

$$= P_{ij}^{(K_2)} - P_{ij}^{(K_1)}$$

Will not change Performance

(2) $$\tilde{P}_{ij}^{(K)} = P_{ij}^{(K)} - \mu_{ij}$$

$$dist_{ij} = \tilde{P}_{ij}^{(K_2)} - \tilde{P}_{ij}^{(K_1)}$$

$$= P_{ij}^{(K_2)} - P_{ij}^{(K_1)}$$

Will not Change Perform

(3) $$\tilde{P}_{ij}^{(K)} = \frac{P_{ij}^{(K)} - \mu}{\sigma}$$

Will not change Perform

$$dist_{ij} = \frac{P_{ij}^{(K_2)} - P_{ij}^{(K_1)}}{\sigma} \longrightarrow$$

$$\boxed{dist = \frac{1}{\sigma} \sum_{ij} \left| P_{ij}^{(K_2)} - P_{ij}^{(K_1)} \right|}$$

(4) $$\tilde{p}_{ij}^{(k)} = \frac{p_{ij}^{(k)} - \mu_{ij}}{\sigma_{ij}}$$

Will change Performance

$$dist_{ij} = \frac{p_{ij}^{(k_1)} - p_{ij}^{(k_2)}}{\sigma_{ij}}$$

$$\cancel{dist = \sum_{ij} \left(\frac{p_{ij}^{(k_1)} - p_{ij}^{(k_2)}}{\sigma_{ij}} \right)^2}$$

$$dist = \sum_{ij} \left| \frac{p_{ij}^{(k_1)} - p_{ij}^{(k_2)}}{\sigma_{ij}} \right|$$

(5) Will change Performance

Inline Question 3

1. No
2. Yes
3. No
4. Yes