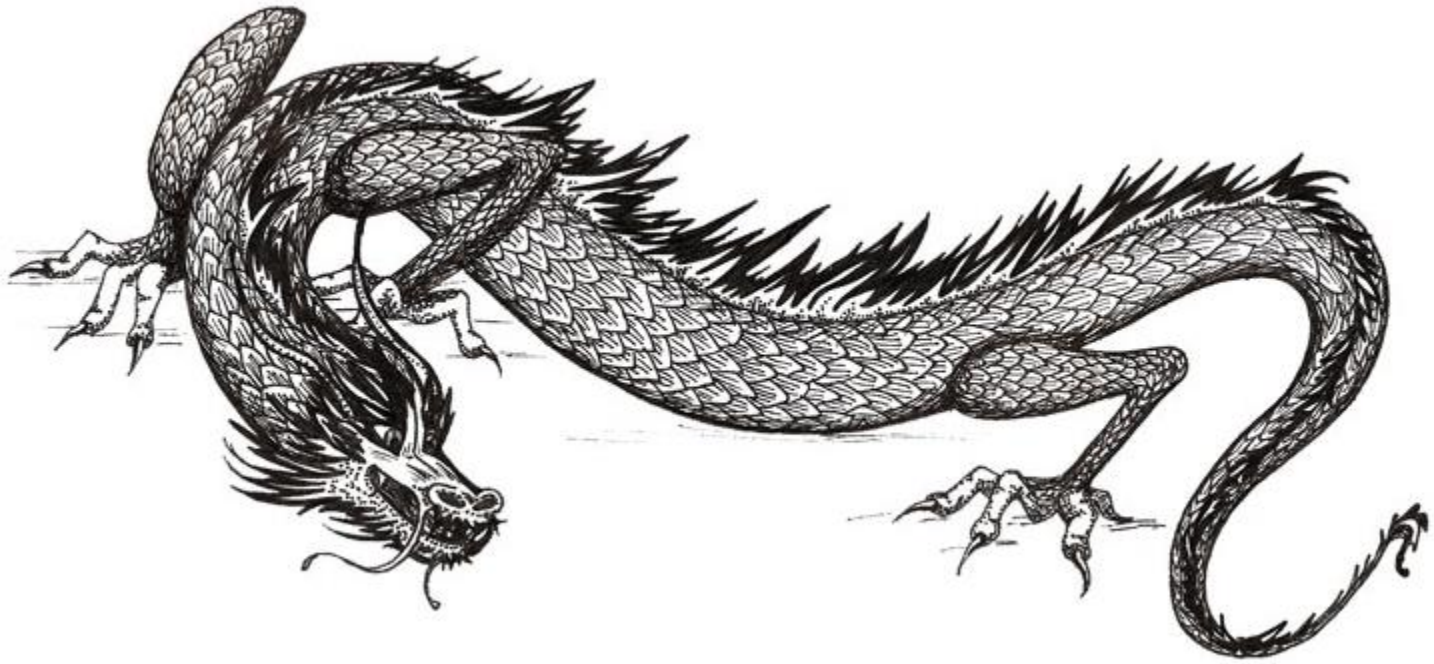


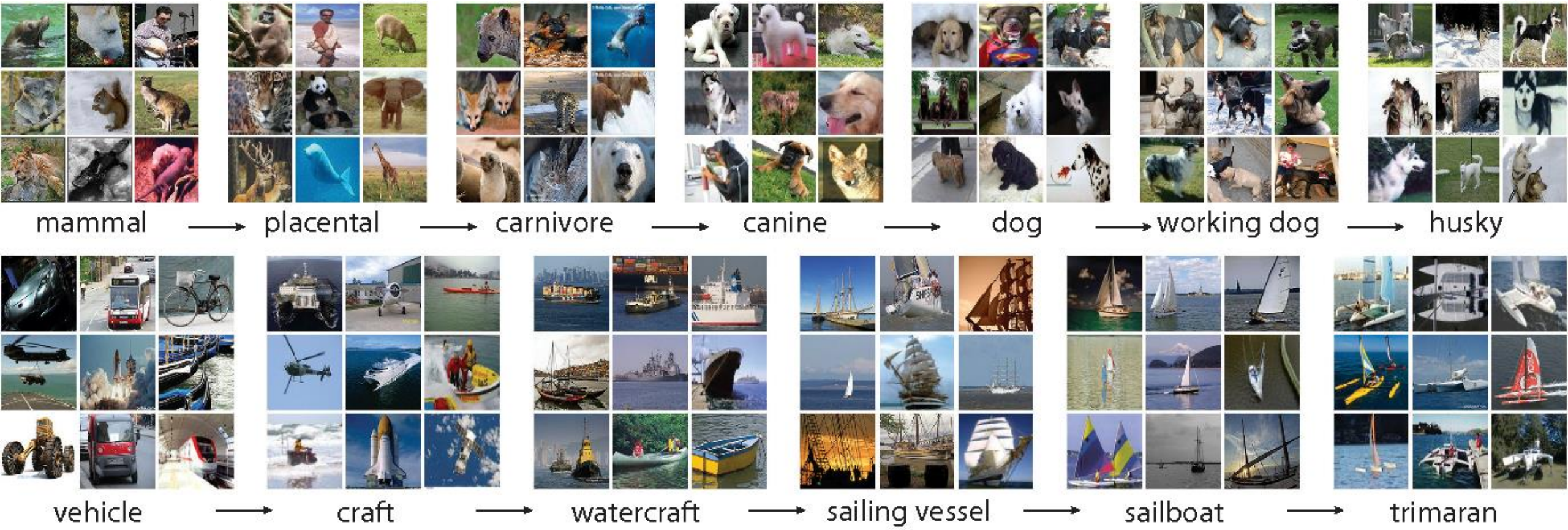
手写数字识别

主讲人：龙良曲

Discrete Prediction

- $y = w * x + b$
- [up, left, down, right]
- [dog, cat, whale, bird, …]

Image Classification

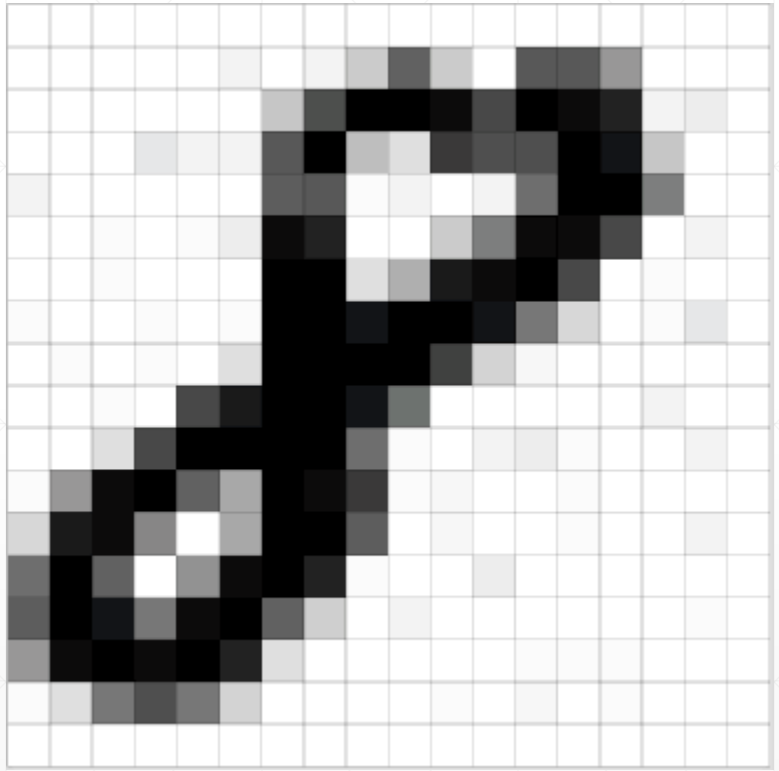


Hand-written Digits Recognition

- MNIST
  - 7000 images per category
  - train/test splitting: 60k vs 10k

Image

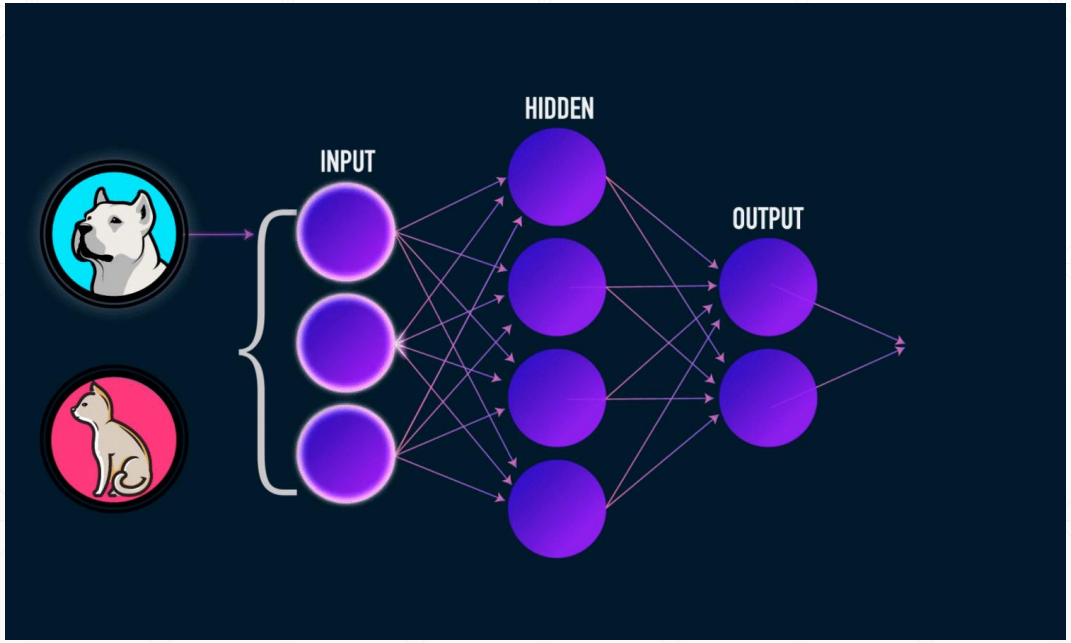
- [28,28,1]
- →[784]



Input and Output

- x: [b, 784]

- prediction:
  - 1. dog = 0, cat = 1, fish = 2, …
  - 2. dog = [1, 0, 0, …]
  - cat = [0, 1, 0, …]
  - fish = [0, 0, 1, …]



https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a

Regression VS Classification

- $y = w * x + b$
  - $y \in R^d$

- $out = X@W + b$
  - out: [0.1, 0.8, 0.02, 0.08]
- $pred = argmax(out)$
  - pred: 1
  - label: 2

Computation Graph

- $out = X@W + b$

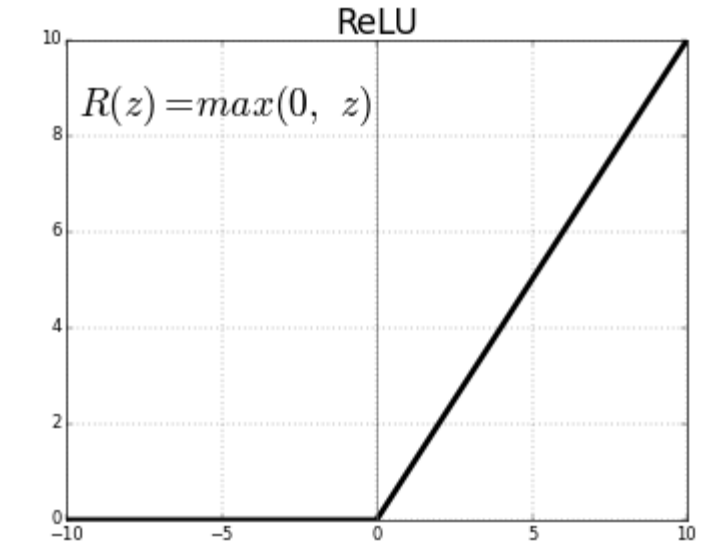
- X: [b, 784]
- W: [784, 10]
- b: [10]
- out: [b, 10]

It's Linear!

- $out = X@W + b$

- $out = \boldsymbol{f}(X@W + b)$

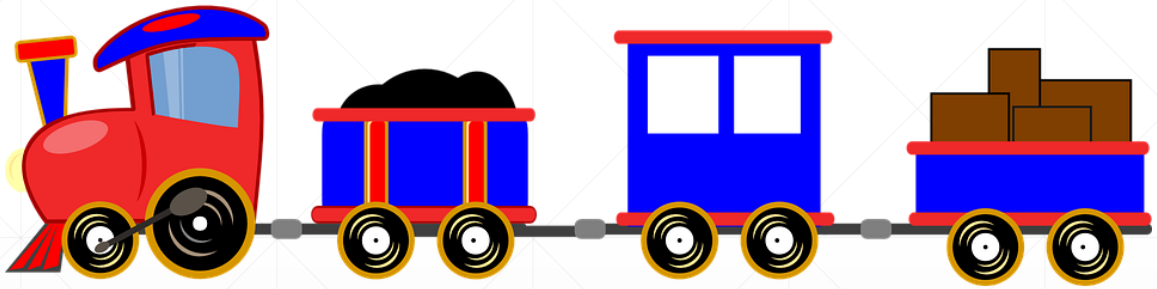
- $out = \boldsymbol{relu}(X@W + b)$



It’s too simple!

- $out = \boldsymbol{relu}(X@W + b)$

- $h_1 = \boldsymbol{relu}(X@W_1 + b_1)$
- $h_2 = \boldsymbol{relu}(h_1@W_2 + b_2)$
- $out = \boldsymbol{relu}(h_2@W_3 + b_3)$



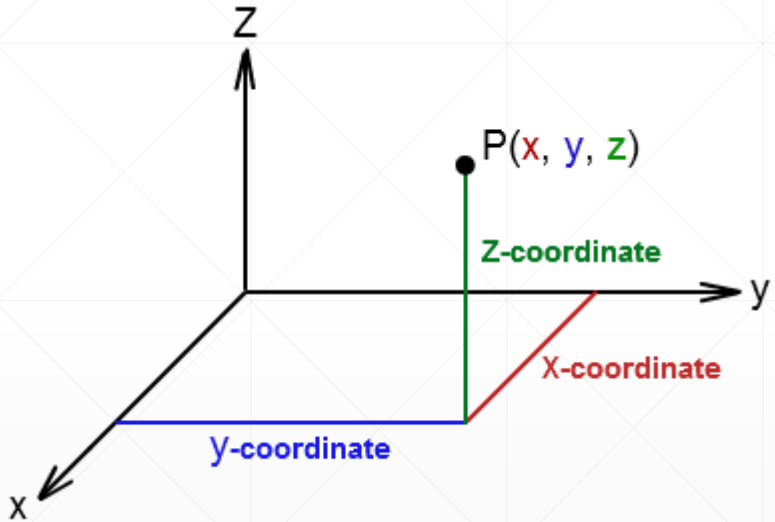
Particularly

- $X = [v1, v2, \ldots, v784]$
  - X: [1, 784]
- $h_1 = \boldsymbol{relu}(X@W_1 + b_1)$
  - W_1 : [784, 512]
  - b_1 : [1, 512]
- $h_2 = \boldsymbol{relu}(h_1@W_2 + b_2)$
  - W_2 : [512, 256]
  - b_2 : [256]
- $out = \boldsymbol{relu}(h_2@W_3 + b_3)$
  - W_3 : [256, 10]
  - b_3 : [10]

$[0, 0, 0.01, 0.1, 0.8, 0, \ldots]$

Loss?

- out: [1, 10]
- Y/label: 0~9
  - eg.: 1 → [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
  - eg.: 3 → [0, 0, 0, 3, 0, 0, 0, 0, 0, 0]
- Euclidean Distance: out → Label
  - MSE



In a nutshell

- $out = relu\{relu\{relu[X@W_1 + b_1]@W_2 + b_2\}@W_3 + b_3\}$

- $pred = argmax(out)$
- $loss = MSE(out, label)$

- minimize $loss$
  - $[W_1', b_1', W_2', b_2', W_3', b_3']$

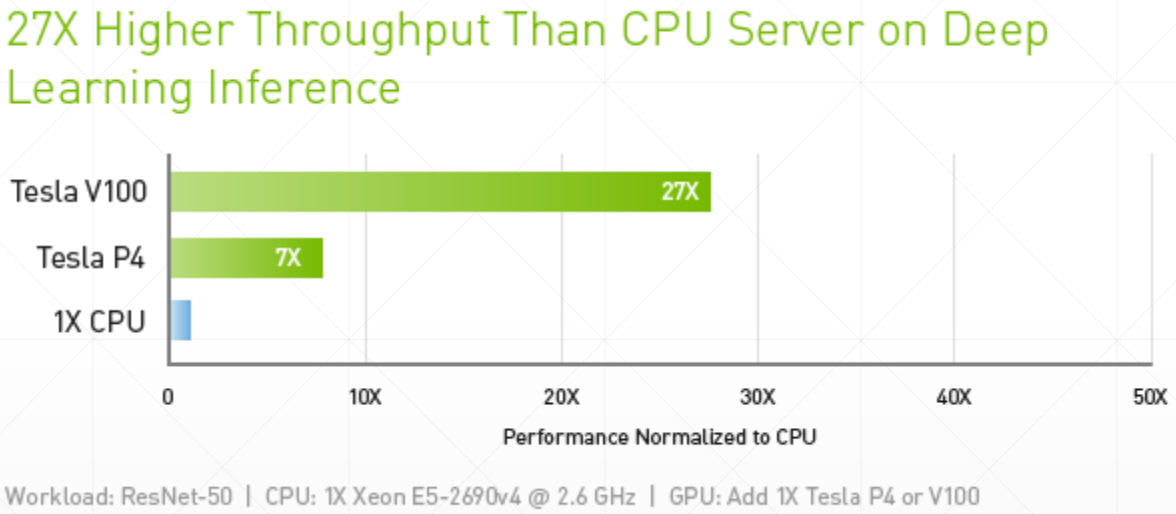
Deep Learning?

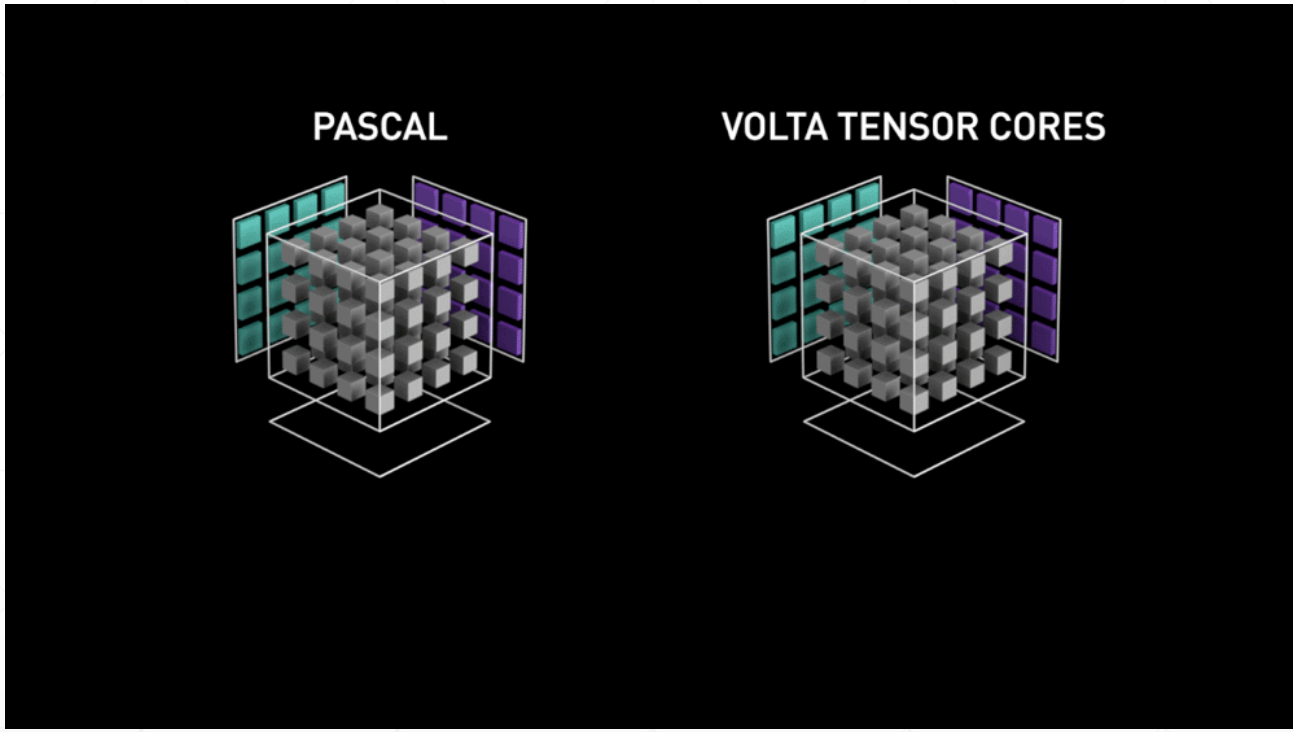
- We have not seen it.
- But we already master it.
- We will show you It’s (almost) Deep Learning!

Classification Procedure

- Step1. Compute $[h_1, h_2, out]$
- Step2. Compute $Loss$
- Step3. Compute gradient and update $[W_1', b_1', W_2', b_2', W_3', b_3']$
- Step4. Loop

We need TensorFlow





https://www.nvidia.com/en-au/data-center/tensorcore/

Next

- Step1. Have fun on MNIST classification
- Step2. and we learn TensorFlow
- Step3. and we implement Step1. by ourselves!

下一课时

Enjoy MNIST!

Thank You.