.NET Conf

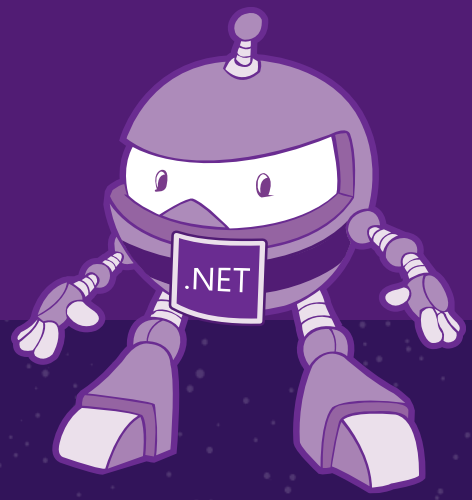
探索 .NET 新世界

Host by

STUDY4

基於Onnx Runtime 深度學習應用開發

尹相志 Allan Yiin



什麼是ONNX

(open format to represent deep learning models)

.NET Conf

Frameworks

Caffe2

Chainer

Cognitive Toolkit

mxnet

PyTorch

PaddlePaddle

MATLAB

sas

Neural Network Libraries

Converters

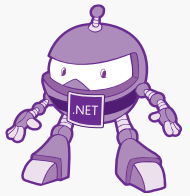
K

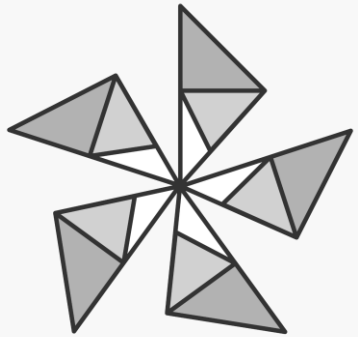
ML

scikit learn

dmlc XGBoost

LibSVM





ONNX RUNTIME

ONNX Runtime又是什麼?

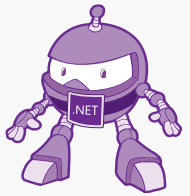
https://github.com/microsoft/onnxruntime

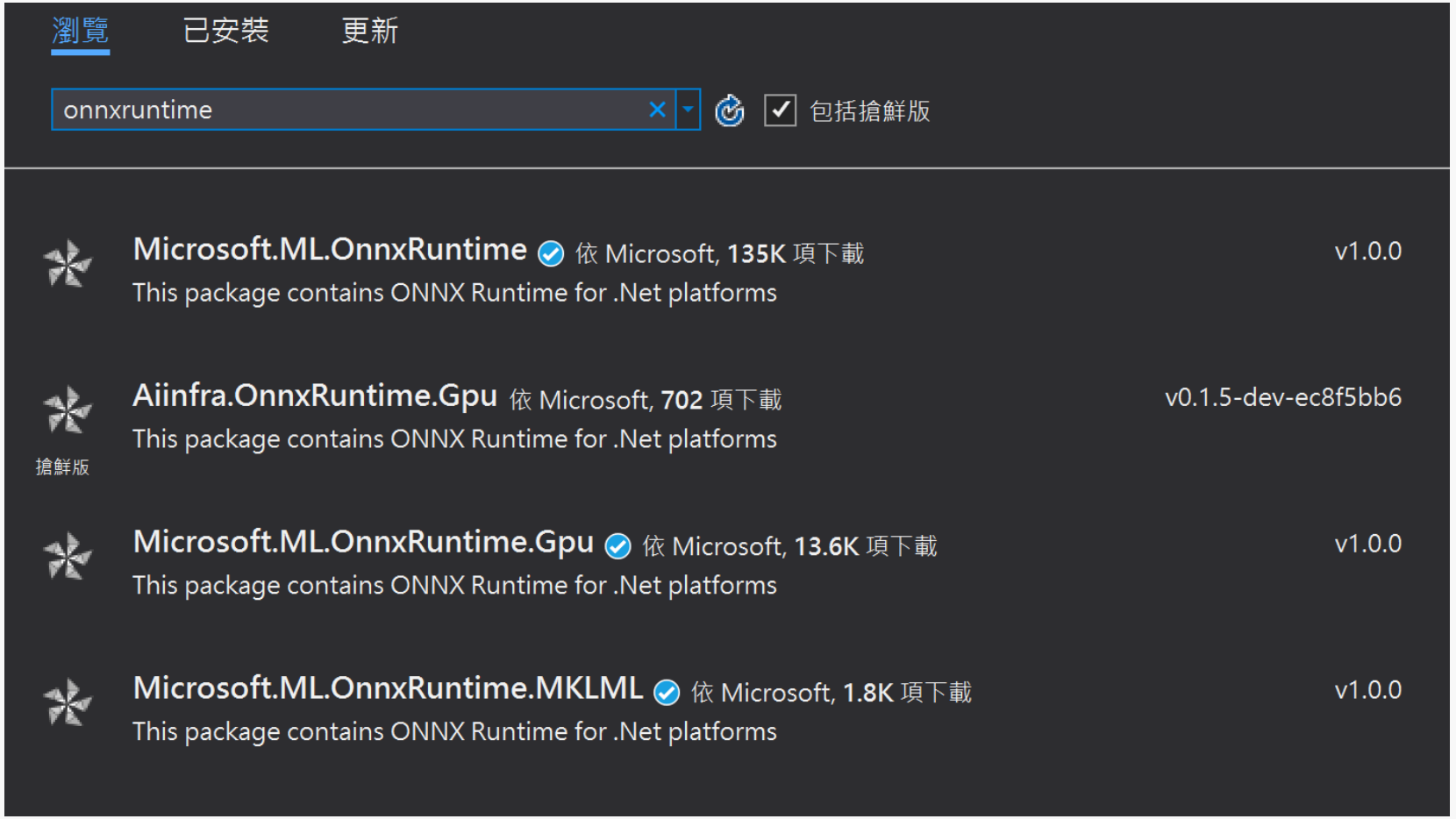
可以支持的語言

- Python (3.5~3.7)
- C
- C# (.Net Standard>1.1)
- C++
- Ruby

可以支援以下加速機制

- MLAS (Microsoft Linear Algebra Subprograms)
- NVIDIA CUDA (**CUDA 10.0** and **cuDNN 7.6**)
- Intel MKL-ML
- Intel MKL-DNN - subgraph optimization
- Intel nGraph
- NVIDIA TensorRT
- Intel OpenVINO
- Nuphar Model Compiler
- DirectML
- ACL (in preview, for ARM Compute Library)





我該如何獲得ONNX模型呢?

CNTK

```
nodes=C.logging.get_node_outputs(z)
model1=combine([nodes[63].owner])
model1.save('vae_model.onnx',ModelFormat.ONNX)
```

Pytorch

```
pnet, rnet, onet = load_net(args, 'pnet'), load_net(args, 'rnet'), load_net(args, 'onet')
pnet.to(device)
torch.onnx.export(pnet, torch.randn(1, 3, 12, 12, device='cuda'), "pnet.onnx", verbose=True, input_names=[ "input_1" ],output_names=[ "output1" ])
rnet.to(device)
torch.onnx.export(rnet, torch.randn(1, 3, 24, 24, device='cuda'), "rnet.onnx", verbose=True,input_names=["input_1"], output_names=["output1"])
onet.to(device)
torch.onnx.export(onet, torch.randn(1, 3,48, 48, device='cuda'), "onet.onnx", verbose=True,input_names=["input_1"], output_names=["output1"])
```

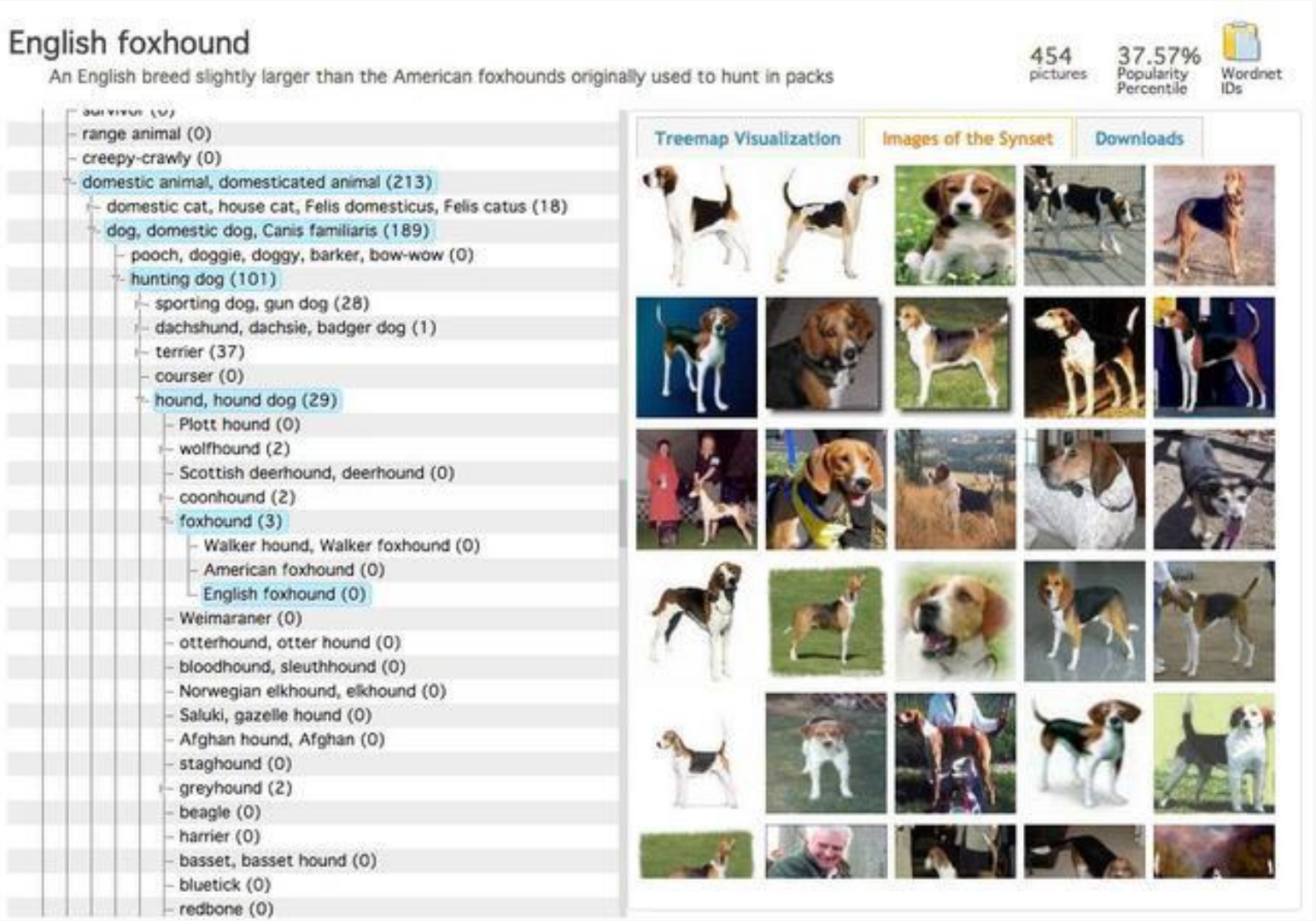
...那微軟的新歡Tensorflow 呢....

```
model = MobileNetV2(alpha=1.0, include_top=True,weights='imagenet',pooling=None,classes=1000)

import keras2onnx
onnx_model = keras2onnx.convert_keras(model, model.name)
import onnx
onnx.save_model(onnx_model, 'mobilenet_v2.onnx')
```

很抱歉囉，不支持tf 2.0..... (要怪該怪誰呢?)

2012年
深度學習技術
首次參賽IMAGENET
將歷史錯誤率降到新低

IMAGENET

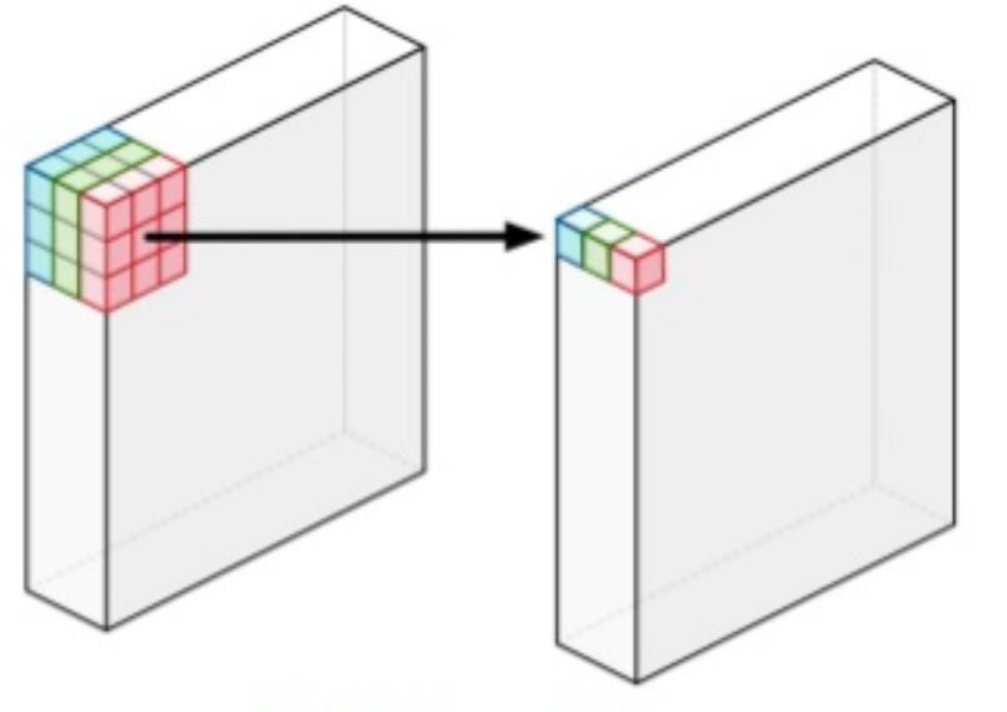


最早從100萬張圖片(1000個類別)，擴充至1500萬張圖片，2.2萬個類別

MobileNet V2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Table 2: MobileNetV2 : Each line describes a sequence of 1 or more identical (modulo stride) layers, repeated n times. All layers in the same sequence have the same number c of output channels. The first layer of each sequence has a stride s and all others use stride 1. All spatial convolutions use 3×3 kernels. The expansion factor t is always applied to the input size as described in Table 1.



深度可分離卷積

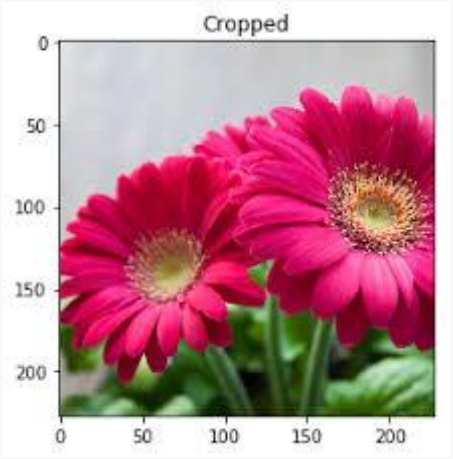
先看Python版

```python
import numpy as np
import onnxruntime
import cv2

#labels=open('imagenet_labels.txt',encoding='utf-8-sig').readlines()

with open('imagenet_labels.txt', 'r') as f:
    labels = [l.rstrip() for l in f]

img=cv2.imread('dog.jpg') #opencv讀入形狀為 HWC  (224,224,3) 像素排列為BGR
img=img.transpose([2,0,1])/255. #把形狀調整成CHW

mean = np.expand_dims(np.expand_dims(np.array([0.485, 0.456, 0.406]),-1),-1)
std =  np.expand_dims(np.expand_dims(np.array([0.229, 0.224, 0.225]),-1),-1)
img=(img-mean)/std #依照平均值與標準差正規化

imgf=np.reshape(img,-1)

img=np.expand_dims(img,0).astype(np.float32)  #把形狀變成(1,3,224,224)
```

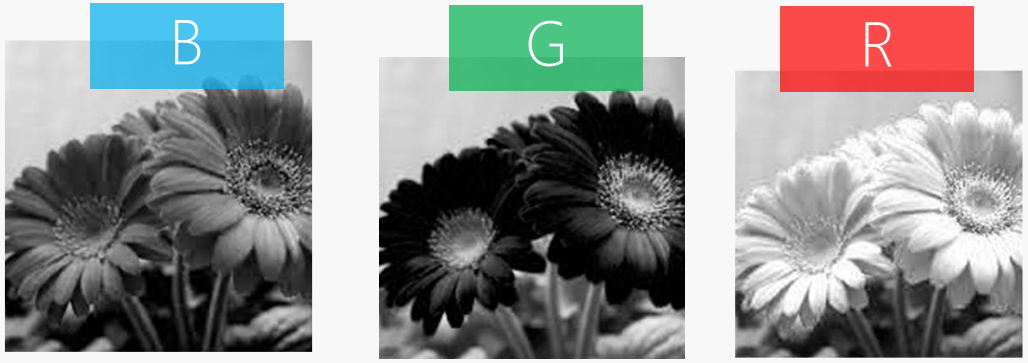
```python
sess = onnxruntime.InferenceSession('../Models/mobilenetv2-1.0.onnx')

input_shape=sess.get_inputs()[0].shape
input_name = sess.get_inputs()[0].name
output_name = sess.get_outputs()[0].name
pred_onnx =sess.run ([output_name], {input_name:img})[0]

pred_onnx = np.squeeze(pred_onnx)
pred_onnx=np.exp(pred_onnx)/np.sum(np.exp(pred_onnx))
prob = list(np.argsort(pred_onnx)[::-1][:5])

print('\n'.join(['{0}  {1:.3%}'.format(labels[item],pred_onnx[item]) for item in prob]))
```

```
C:\Anaconda3\python.exe C:/Users/Allan/source/repos/DataDecision.onnx/OnnxDemo_py/onnx_mobilenet.py
n02113186 Cardigan, Cardigan Welsh corgi  58.042%
n02113023 Pembroke, Pembroke Welsh corgi  14.762%
n02109961 Eskimo dog, husky  8.807%
n02091467 Norwegian elkhound, elkhound  3.302%
n04409515 tennis ball  2.575%
```

色彩的通道 (CHW? HWC)



人類習慣RGB

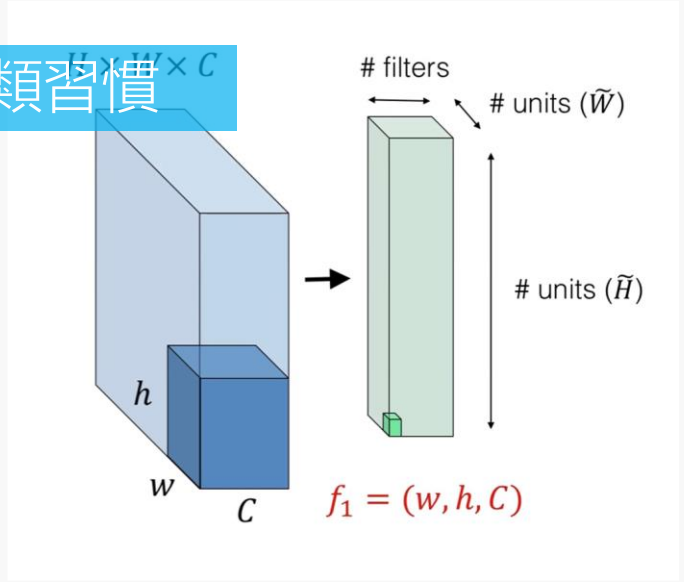


OpenCV支援

CuDNN支援

特徵圖通常被放置在第三維度
或者是另外擴充至第四維度

人類習慣



如何表達不同的類別?
最簡單粗暴的做法:

One Hot

如何生成One Hot向量
oh=np.zeros(維度數)
oh[案例]=1

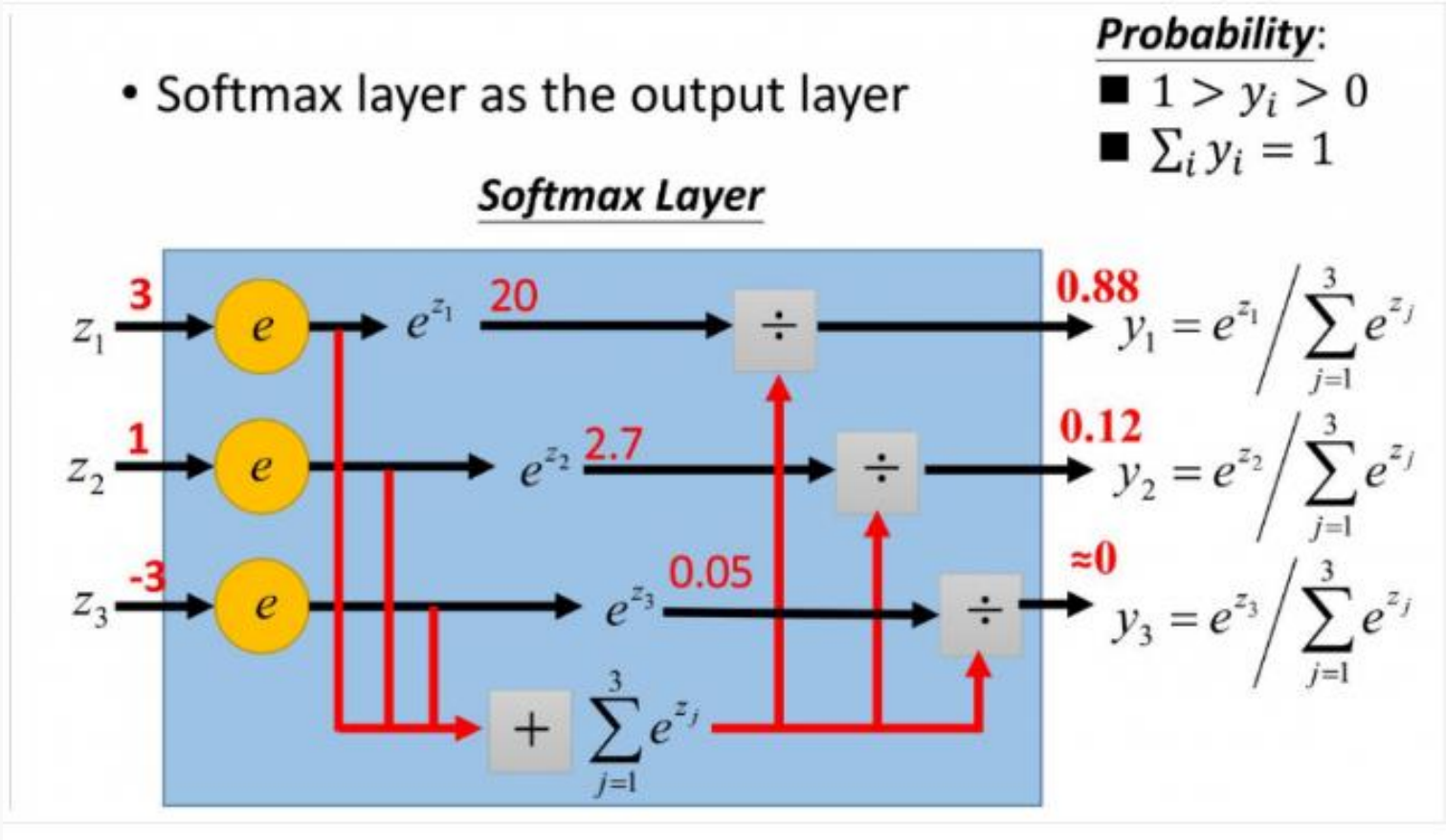
如何找出最大機率可能性
argmax

#	Color
0	Red
1	Green
2	Blue
3	Red
4	Blue

#	Red	Green	Blue
0	1	0	0
1	0	1	0
2	0	0	1
3	1	0	0
4	0	0	1

$$S_i = \frac{e^{V_i}}{\sum_j e^{V_j}}$$

softmax



放心! 當然還有.net的版本

幾個需要解決的問題

.net沒有numpy，目前缺乏高效的tensor運算方案
如何確認是否環境裡有可用GPU?
如何定義輸入與輸出

幾個常見的坑

cntk與pytorch是**CHW**排列，像素順序為**BGR**
tensorflow是**HWC**排列，像素順序為**RGB**
cntk與pytorch傾向像素除以**255**後再減去**平均值**除以**標準差**
tensorflow則是減**127.5**除以**127.5**

```
nnx                                    DataDecision.onnx.ImageHelper                    ParallelExtractCHW(Bitmap image, PixelNormalizationMode mode =
    int imageHeight = image.Height;
    var features = new float[imageWidth * imageHeight * 3];

    var bitmapData = image.LockBits(new System.Drawing.Rectangle(0, 0, imageWidth, imageHeight), ImageLockMode.ReadOnly, image.PixelFormat);

    IntPtr ptr = bitmapData.Scan0;

    int bytes = Math.Abs(bitmapData.Stride) * bitmapData.Height;

    byte[] rgbValues = new byte[bytes];
    int stride = bitmapData.Stride;
    // 將RGB值複製到array
    Marshal.Copy(ptr, rgbValues, 0, bytes);
    // 根據pixel format對應像素
    Func<int, int, int, int> mapPixel = GetPixelMapper(image.PixelFormat, stride);
      Parallel.For(0, imageHeight, (int h) =>
      {
        Parallel.For(0, imageWidth, (int w) =>
        {
          Parallel.For(0, 3, (int c) =>
          {

            if (mode == PixelNormalizationMode.ZeroBased)...
            else if (mode == PixelNormalizationMode.ZeroCentral)...
            else if (mode == PixelNormalizationMode.imagenet){
              //[0.485, 0.456, 0.406]
              //[0.229, 0.224, 0.225]
              if (c == 0)
              {
                features[channelStride * c + imageWidth * h + w] =((float)(rgbValues[mapPixel(h, w, c)] / 255f)- 0.485f)/ 0.229f;
              }
              else if (c == 1)
              {
                features[channelStride * c + imageWidth * h + w] = ((float)(rgbValues[mapPixel(h, w, c)] / 255f) - 0.456f) / 0.224f;
              }
```

讀取像素
並且依照pytorch格式
將像素正歸化
為一維陣列

初始化Inference Session的做法非常簡單且高效

```
2 個參考
public static void SettingSession(string model_path)
{

    options.GraphOptimizationLevel= GraphOptimizationLevel.ORT_ENABLE_BASIC;

    try
    {
        session = new InferenceSession(model_path, SessionOptions.MakeSessionOptionWithCudaProvider(0));
    }
    catch (Exception e)
    {
        session = new InferenceSession(model_path, options);
    }
    inputName = session.InputMetadata.Keys.ToList()[0];


}
```

抓取GPU

將輸入數據(float[])打包成 Microsoft.ML.OnnxRuntime.Tensors.DenseTensor<float> 再添加輸入變數名稱打包為 NamedOnnxValue.CreateFromTensor<float>

```
1 個參考
public static NamedOnnxValue BitmapToTensor(Bitmap img)
{
    float[]data=img.ParallelExtractCHW(PixelNormalizationMode.imagenet).ToArray();
    var tensor = new Microsoft.ML.OnnxRuntime.Tensors.DenseTensor<float>(data, InferHelper.session.InputMetadata[InferHelper.inputName].Dimensions);
    return NamedOnnxValue.CreateFromTensor<float>(InferHelper.inputName, tensor);
}
```

如果頭開始有點昏了!?

別擔心，老師有法寶.....

```
string taskOption = Console.ReadKey().KeyChar.ToString();

if (taskOption == "0")
{
    try
    {
        DateTime d = DateTime.Now;
        Console.WriteLine("載入模型");
        InferHelper.SettingSession("Models/mobilenetv2-1.0.onnx");
        Console.WriteLine("輸入圖片為dog.jpg");
        var result = InferHelper.InferImagenet(new Bitmap("Images/dog.jpg")).ToList()[0];
        var probs = result.AsEnumerable<float>().ToList();
        probs = probs.Select(x => (float)Math.Exp(x)).ToList();
        float sum_probs = probs.Sum();
        probs = probs.Select(x => x / sum_probs).ToList();
        var maxidx = probs.ArgMax(x => x);
        Console.WriteLine(labels[maxidx]);
        Console.WriteLine(string.Format("機率為:{0:p3}", probs[maxidx]));

    }
```

```
C:\Users\Allan\source\repos\DataDecision.onnx\ConsoleApp
請問您想測試的是:
0.  測試imagenet
0載入模型
輸入圖片為dog.jpg
n02113186 Cardigan, Cardigan Welsh corgi
機率為:58.925%
```

代碼哪裡拿?

https://github.com/AllanYiin/DataDecision.onnx

模型怎麼做?

https://github.com/AllanYiin/DeepBelief_Course4_Examples

特別感謝

Microsoft® MVP Most Valuable Professional

SEA 台灣軟體工程學會 Taiwan

R-Ladies Taipei

TAIWAN

91APP

多奇·數位創意

Wishing-Soft

KKTIX

Geek & Nerd Studios

HackMD

以及各位參與活動的你們

STUDY4.TW 為學習而生

.NET Conf