2024 鐵人賽 – 我數學就爛要怎麼來學 DNN 模型安全

Day 22 – Deep Leakage from Gradients

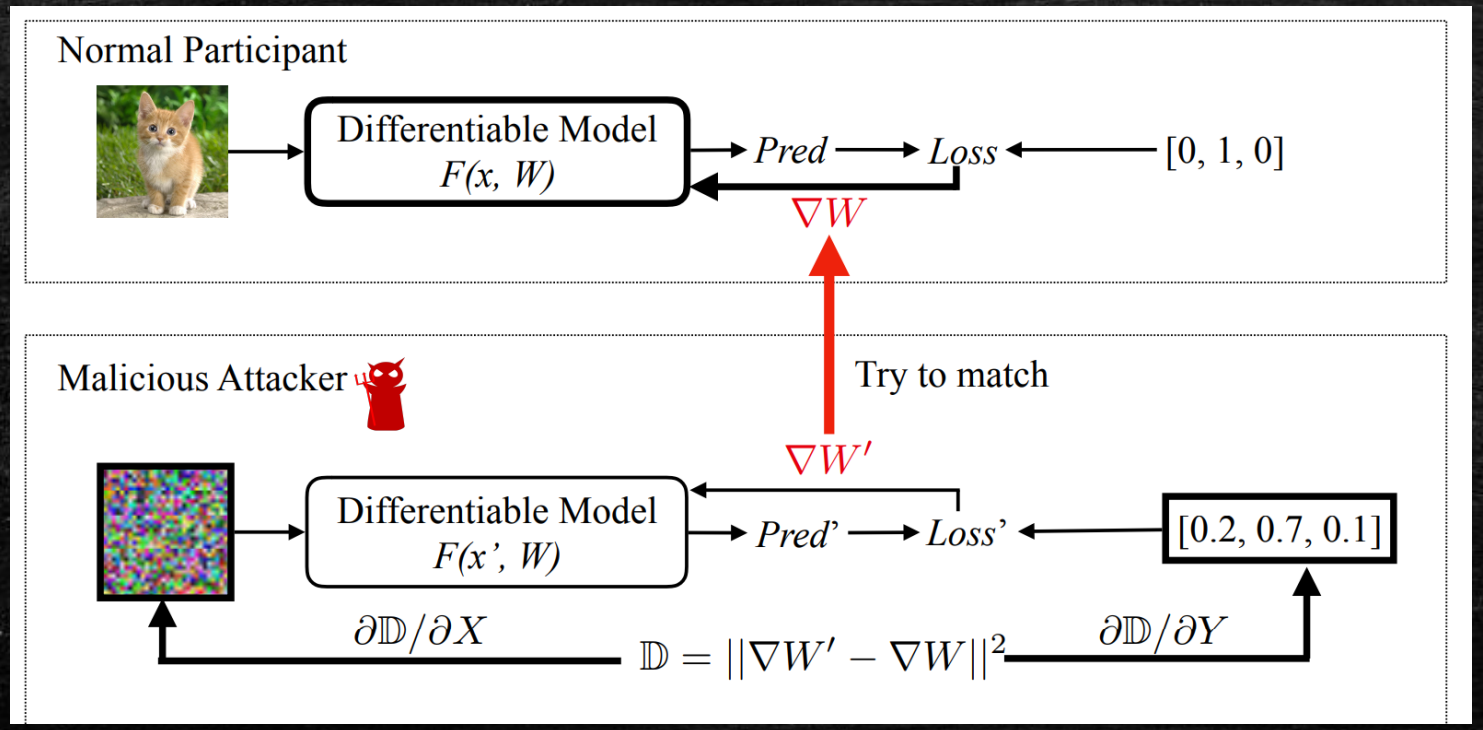
大綱

- Deep Leakage from Gradients
  - 前情提要
  - Hint : Model subclassing
  - 程式實作
- 結論

前情提要

Figure 2: The overview of our DLG algorithm. Variables to be updated are marked with a bold border. While normal participants calculate ∇W to update parameter using its private training data, the malicious attacker updates its dummy inputs and labels to minimize the gradients distance. When the optimization finishes, the evil user is able to obtain the training set from honest participants.

- Deep Leakage from Gradients (2019)
  - 簡而言之，當製造出一組資料及標籤的梯度跟得到的一致時，這組資料及標籤就接近當初的輸入資料



https://arxiv.org/pdf/1906.08935

程式實作 – 參考資料 - 神之一手模型

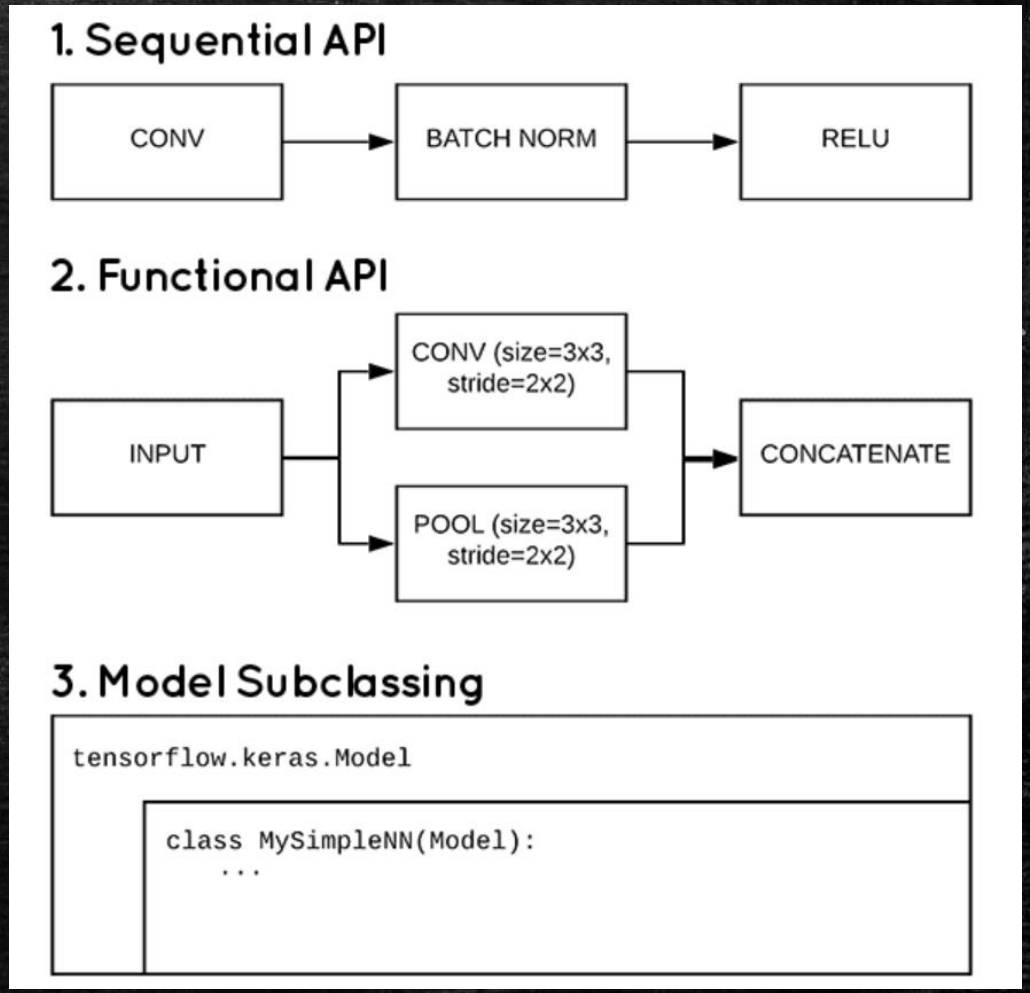
- 這個演算法不好做，因為 loss function 變成輸入資料、標籤得到的梯度差異，然後更新資料變成回去更新輸入資料跟標籤
- 回想一下以前的模型的情況是 loss function 是預測值跟實際值的差異，然後更新資料是更新模型的權重資料
- 所以，要想辦法把輸入資料，標籤數值搞成模型權重去做更新，想法會很有趣

https://medium.com/@EastGeno/deep-leakage-from-gradients-%E5%BE%9E%E6%A2%AF%E5%BA%A6%E6%8B%BF%E5%88%B0%E4%BD%A0%E7%9A%84%E8%B3%87%E6%96%99-d23232c03bd2

Hint : Model subclassing

- Keras 的模型有三種建構方式
  - Sequential API
  - Functional API
  - Model subclassing

- Model subclassing 的彈性最高
  - 優點在於可以在模型內插入程式
  - 缺點則是不容易看出模型結構



Hint : Model subclassing

https://www.tensorflow.org/guide/keras/making_new_layers_and_models_via_subclassing

```
class ResNet(keras.Model):

    def __init__(self, num_classes=1000):
        super().__init__()
        self.block_1 = ResNetBlock()
        self.block_2 = ResNetBlock()
        self.global_pool = layers.GlobalAveragePooling2D()
        self.classifier = Dense(num_classes)

    def call(self, inputs):
        x = self.block_1(inputs)
        x = self.block_2(x)
        x = self.global_pool(x)
        return self.classifier(x)


resnet = ResNet()
dataset = ...
resnet.fit(dataset, epochs=10)
resnet.save(filepath.keras)
```

程式實作

```
# 定應屬於自己的 base model
class myModule(tf.keras.Model):
    def __init__(self) :
        super(myModule,self).__init__()
        # 這邊只做一層是因為做兩層的回覆效果就不好了
        #self.dense1 = Dense(128,activation=tf.nn.sigmoid)
        self.dense2 = Dense(10,activation=tf.nn.softmax)

    def call(self, inputs, training=True) :
        #x = self.dense1(inputs)
        #x = self.dense2(x)
        x = self.dense2(inputs)
        return x

    # 這邊在模型內定義一個 function, 傳入輸入資料跟標籤會回傳當下 gradient
    def gradient(self,x,y) :
        with tf.GradientTape() as tape:
            predict = self(x)
            tape.watch(self.weights)
            loss = tf.reduce_mean(tf.losses.categorical_crossentropy(predict, y))
            g = tape.gradient(loss, self.weights)

        # 以下只是把梯度所有資料串接成一個很長的 1 x N 陣列
        gradient_all = tf.reshape(g[0], (1,-1))
        for grad in g[1:]:
            gradient_all = tf.concat([gradient_all, tf.reshape(grad, (1,-1))], axis=-1)
        return gradient_all


my_model = myModule()
```

程式實作

https://stackoverflow.com/questions/61427583/how-do-i-plot-a-keras-tensorflow-subclassing-api-model

結論

- 回憶一下 Keras 建立模型三本柱 - Sequential API、Functional API、Model subclassing 在這次系列賽過程中都練習一遍了

- 有興趣的話可以參考 Coursera - Custom Models, Layers, and Loss Functions with TensorFlow
- https://www.coursera.org/learn/custom-models-layers-loss-functions-with-tensorflow