Introduction to Tensorflow for NFL Sports Analytics

Pranav Tavildar

UConn Sports Analytics Symposium 2022

October 8, 2022

Table of contents

What is Tensorflow?

Open Source Deep Learning Library that makes creating Models more:

- Easy
- Flexible
- Fast
- Reproducable

The History of Tensorflow

Evolution of Tensorflow

- Product of Google Brain Team
- Evolution from Distbelief (2011)
- Possible because of Tensor Processing Unit

Tensorflow vs Alternatives

Tensorflow

- Preferred by Companies
- Easily Deployable and Versatile

Pytorch

- Preferred by Research Labs
- Easily Debuggable

Caffe

- Used by Researchers and Startups
- Useful for Small but Specific and fast use cases

What is a Tensor?

- Tensor is data with dimension.
- 0-d tensor: scalar

$$c = 5$$

- 1-d tensor: vector

$$\mathrm{c} = \begin{pmatrix} 1 \\ \vdots \\ 5 \end{pmatrix}$$

- 2-d tensor: matrix

$$\mathrm{c} = \begin{pmatrix} 1 & \cdots & 5 \\ \vdots & \ddots & \vdots \\ 5 & \cdots & 5 \end{pmatrix}$$

Why Use Tensors?

- Vector, matrix operations and gradient (derivative) are dominant in machine learning and deep learning.
- GPU structure leads to a powerful ability to solve linear tensor operations.

How to install Tensorflow Locally

- Anaconda management (GUI): click "environment" –> choose "Not installed" –> search "tensorflow"
- Anaconda Prompt:

```
conda install tensorflow
```

- Pip:

```
pip install tensorflow
```

- Verify whether your installation is successful: (open jupyter notebook or run at .py file)

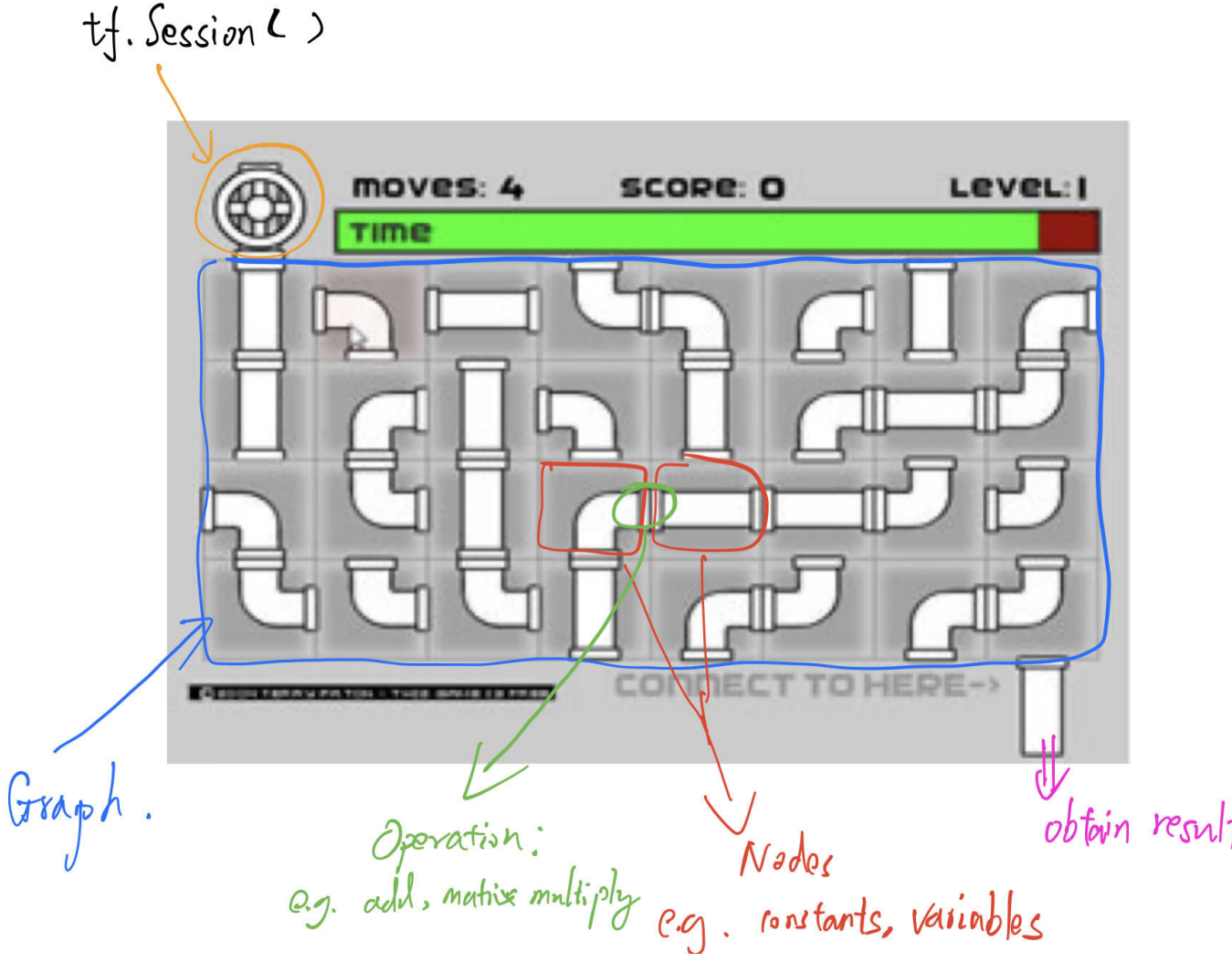
```
1 import tensorflow as tf
2 import tensorflow.compat.v1 as tfv1
3 tf.__version__
```

3 key concepts in tensorflow programming

- Operations: Data operations like "matmul" (matrix multiplication), "add".
- Graph: Build GRAPH which represents the data flow of the computation.
- Session: Run SESSION which executes the operations on the graph.

3 key concepts in tensorflow programming

- Operations: Da multiplication),
- Graph: Build G computation.
- Session: Run S graph.



Tensorflow V1 versus V2

- V1: Nodes $\rightarrow$ Operations $\rightarrow$ Graph $\rightarrow$ Session $\rightarrow$ import the data
- V2: import the data $\rightarrow$ Nodes $\rightarrow$ Operations (eager execution)

Pros and Cons:

- V1 Pros: Integrity, a big picture of view
- V1 Cons: Lack of flexibility
- V2 Pros: Flexibility, you can always see the output step by step
- V2 Cons: Further efforts are frequently needed to pack the code up into a final package or class function

Therefore, V1 is frequently used in final delivery but V2 is good for debugging.

Tensorflow V1 versus V2

Good news: V1 are V2 are both be accessible by tensorflow package.

- Default is V2 (import tensorflow as tf)
- Use "import tensorflow.compat.v1 as tfv1" to call the functions and methods under V1

Basic widget: Constant

The constant can be either scalar, vector, or matrix. It is just a concept which is opposite with variable. Variable can be changed in the future by using tf.assign() method, while the constant can not.

```
1 s = tf.constant(2)
2 m = tf.constant([[1,2],[3,4]])
3 m = tf.constant([1,2,3,4],shape=[2,2])
```

With these two node, we can conduct a GRAPH and SESSION example. Here we use "tensorboard" for GRAPH and tf.Session() function for SESSION.

A simplest graph and session

```
g = tf.Graph()
with g.as_default():
    s = tf.constant(2)
    m = tf.constant([[1,2],[3,4]])
    mmul = s*m
g
```

Here, we have already completed a graph, to "open the faucet", we use tf.Session(), like:

```
with tfv1.Session(graph=g) as sess:
    print(sess.run(mmul))
```

Finally, we get the result. Remark: The tf.Session() only exist in tensorflow v1, so we call this function under v1 platform.

Basic widget: Variable

Variables are used to hold and update parameters. Another important difference with the constant is that it be treated as variable in the calculation of gradient. To create a variable,

```
1  w = tf.Variable(tf.ones((2,2))) # 2*2 matrix with all elements as 1
```

Alert

The variable must be initialized immediately after "the data faucet is open" (i.e. tfv1.Session()).

```
1  with tfv1.Session() as sess:
2      sess.run(tfv1.global_variables_initializer())
3      print(sess.run(w))
```

Basic widget: Variable

As we mentioned earlier, variable can be assigned with a new value.

```
1  with tfv1.Session() as sess:
2      sess.run(tfv1.global_variables_initializer())
3      print(sess.run(w))
4      # Change w to a 2*2 matrix with all elements as 0
5      sess.run(w.assign(tf.zeros((2,2))))
6      print(sess.run(w))
```

Basic widget: Placeholder

We can notice that although variable is more flexible than constant, it still need a initial value. In practice, sometimes the value of a parameter is determined by the actual data. So we need the placeholder to tell the PC, there is a variable, but I don't tell you the value, I will give you the value when I open the data faucet.

```
1  import numpy as np
2  node1 = tfv1.placeholder(tf.float32,shape = [1,2])
3  node2 = tfv1.placeholder(tf.float32,shape = [1,2])
4  w_linear = tf.matmul(node1,w) + node2
5  with tfv1.Session() as sess:
6      sess.run(tfv1.global_variables_initializer())
7      print(sess.run(w))
8      print(sess.run(w_linear,feed_dict={node1:np.matrix([1.0,2.0]),
9          node2:np.matrix([1.0,2.0])}))
```

Basic Operations

Given x and y,

- x+y (element-wise): tf.add(x,y)
- x-y (element-wise): tf.subtract(x,y)
- x*y (element-wise): tf.multiply(x,y)
- x/y (element-wise): tf.divide(x,y)
- x*y (matrix style): tf.matmul(x,y)
- x<y (judgment): tf.less(x,y)
- x>y (judgment): tf.greater(x,y)
- x<=y (judgment): tf.less_equal(x,y)

ANN

One of the greatest strengths of tensorflow is its ability to mimic learning and make predictions using data. This is done through structures known as Neural Networks. In the scope of this workshop, we're only covering Artificial Neural Networks known as ANNs.

ANNs Continued

- Approach like a black box with input and output
- Neurons
- Weights

NFL Play-By-Play Analysis

Now it is time to create our Tensorflow Model! We're using play-by-play data from the NFL 2021-2022 season gathered from `NFLSavant`. Let's try to create a model that predicts whether the next play is going to be a first down given data.

Shameless Promo

Thank you very much for attending! The material for this workshop is accessible here –> `Session Material`.

Join UConn Data Science Club!

Our UConntact –> `Our UConntact`.

Our Discord –> `Discord`.