AIDL: PROJECTS

Session 4 (2019/05/21): Deep dive into Tensorflow III

Exercise I

Linear regression: the forgotten fully connected layer

$$y' = W \cdot x + b$$

$$L = (y' - y)^2$$

Having:

- y': prediction
- y: ground truth
- W: weight
- b: bias
- x: input

Let TF do the magic

Automatic differentiation: definition

“Automatic differentiation or computational differentiation, is a set of techniques to numerically evaluate the derivative of a function specified by a computer program. AD exploits the fact that every computer program, no matter how complicated, executes a sequence of elementary arithmetic operations (addition, subtraction, multiplication, division, etc.) and elementary functions (exp, log, sin, cos, etc.). By applying the chain rule repeatedly to these operations, derivatives of arbitrary order can be computed automatically, accurately to working precision, and using at most a small constant factor more arithmetic operations than the original program.”

Automatic differentiation: optimizers

Tensorflow provide a different number of optimizers (doc) that will compute the backprop for you and optimize your models. A few examples:

- SGD: tf.train.GradientDescentOptimizer
- Adam: tf.train.AdamOptimizer
- RMSProp: tf.train.RMSPropOptimizer

```
loss = ...
optimizer = tf.train.AdamOptimizer(lr=10e-4)
train_step = optimizer.minimize(loss)

sess.run(train_step, feed_dict={...})
```

Questions?