Slide-3-Basic-Models-in-TensorFlow

Created @Apr 8, 2021 10:38 AM

TensorFlow

Recap

- Two phases of computation from tensorflow execution?
- `tf.constant` vs `tf.Variable`?
  - Constant value are stored in the graph defination
  - Sessions allocate memory to store variable values
- `tf.placehlder` and `feed_dict`?
  - Feed values into placeholders with dictionary (feed_dict)

  ⇒ Esay to use but poor performance
- Try to avoid lazy loading!

Linear Regresion

- Target?

  Find a linear relationship btw X and Y to predict Y from X
- Model?
  - Inference: $Y_{pred} = w * X + b$
  - Loss: MSE
- Phase 1: Assemble our graph?
  - Step 1: Read in data (build your own utils for it)
  - Step 2: Create placeholders for inputs and lables (no need to specify shape)
  - Step 3: Create weight and bias using `get_variable`
  - Step 4: Inference function

- Step 5: Specify loss function
- Step 6: Create optimizer

▼ Phase 2: Training the model

- Step 1: Initialize variables
- Step 2: Run optimizer
- Using `feed_dict` to feed data into X and Y
- Write log files using a FileWriter

▼ Some note for the loss

- Using Huber loss ⇒ robust to outliers
- When the different btw predicted value and real value is small ⇒ squared
- if it too large ⇒ take its absolute value

▼ Some note for handling data?

Placeholder:

- processing outside TensorFlow ⇒ easy to do in Python. But usally using single thread ⇒ slow execution

tf.data:

- Instead of doing inference with placeholders and feeding in data later ⇒ do inference directly with data
- Store data in **tf.data.Dataset ⇒ Can r**ead data from file
- **tf.data.Iterator:** Create an iterator to iterate through samples in Dataset
- Can handling data in Tensorflow!

⇒ **tf.data** perform better than **placeholder**

Logistic Regression

▼ dataset?

MNIST: X - 1 d tensor for each image of size 784, Y - lables

▼ Model?

- Inference: $Y_{pred} = softmax(w * X + b)$
- Loss: cross entropy loss

▼ Process data?

- There is no immediate way to convert Python generators to tf.data ⇒ write utils
- Convert it to tf.data ⇒ create datasets
- Create iterator of dataset: `iterator = train_data.make_initializable_iterator()`
- Initialize iterator with the dataset you want

▼ Phase 1 and Phase 2 are the same as Linear Regression