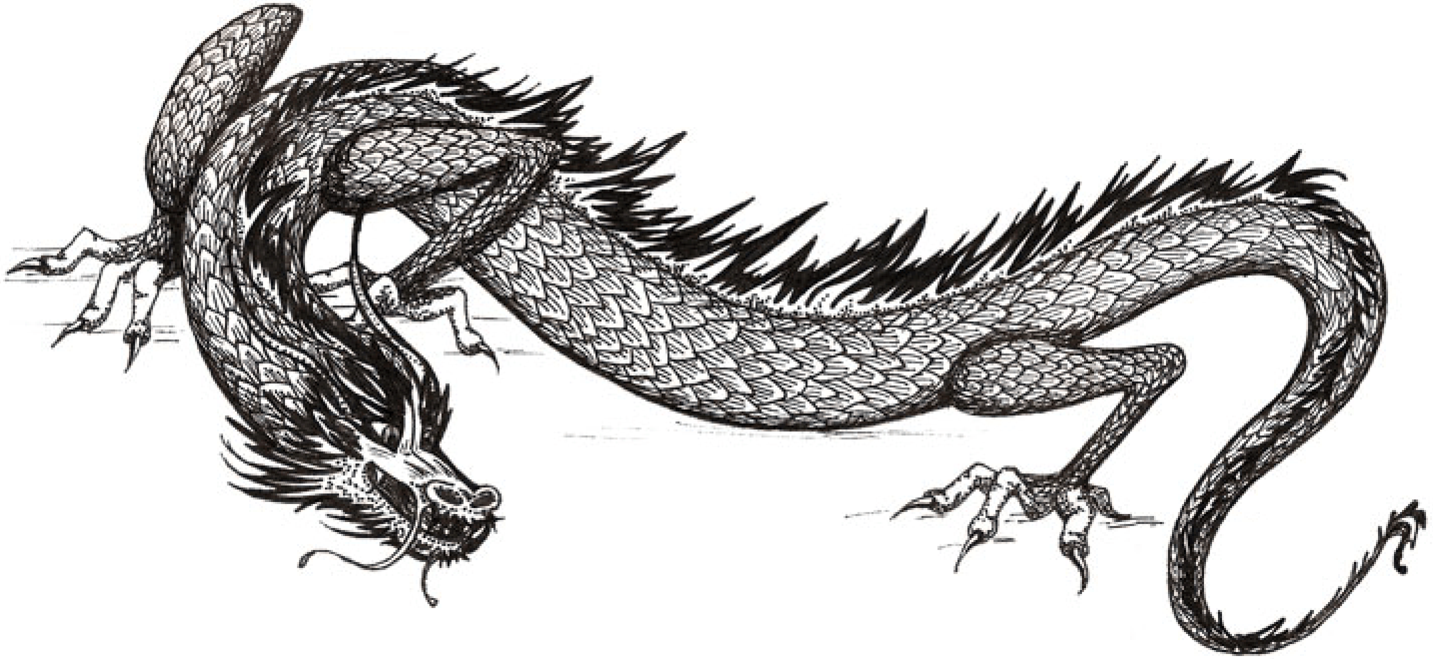


交叉验证

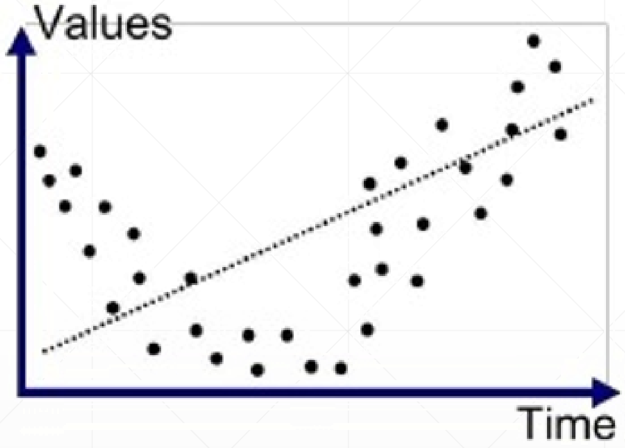
主讲：龙良曲

Outline

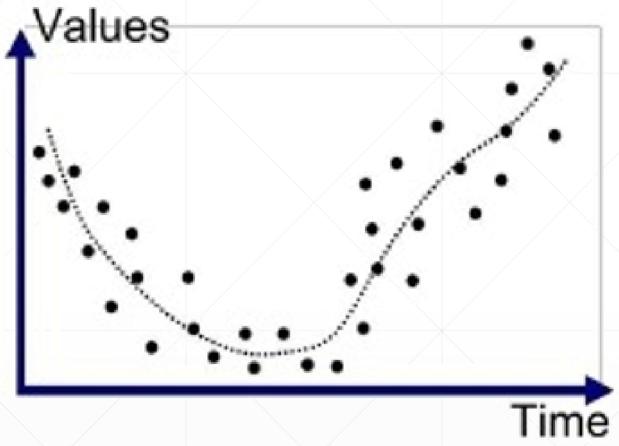
- train/evaluate/test splitting

- k-fold cross-validation

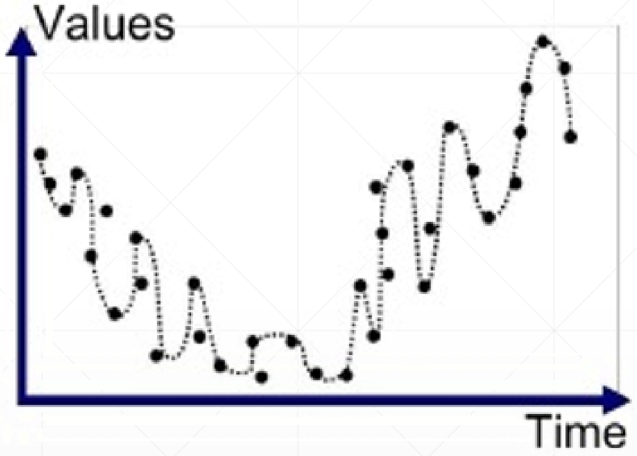
Recap



Underfitted

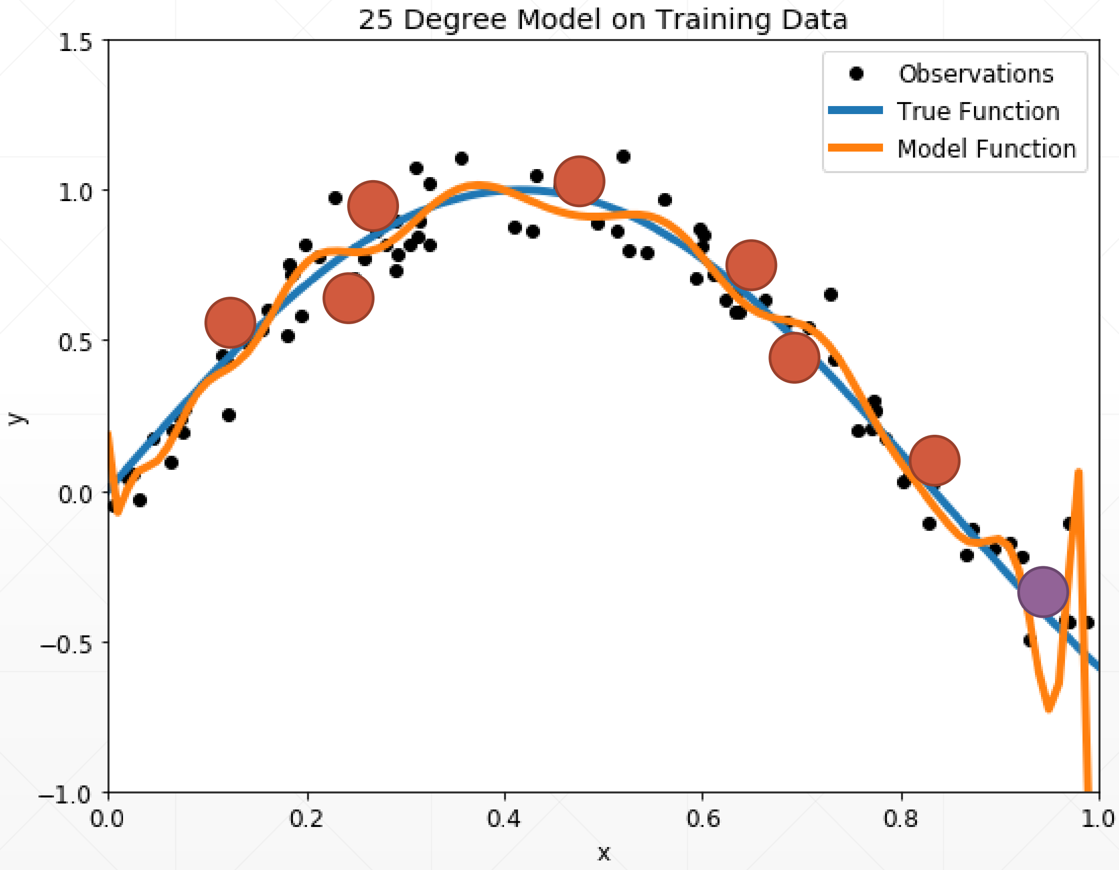


Good Fit/Robust



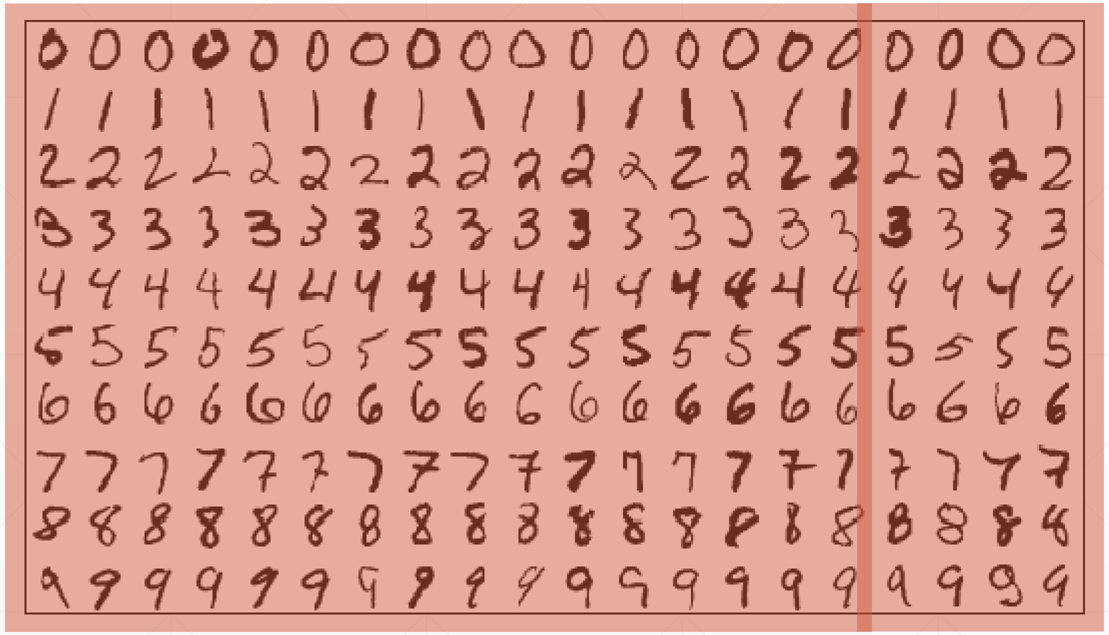
Overfitted

How to detect



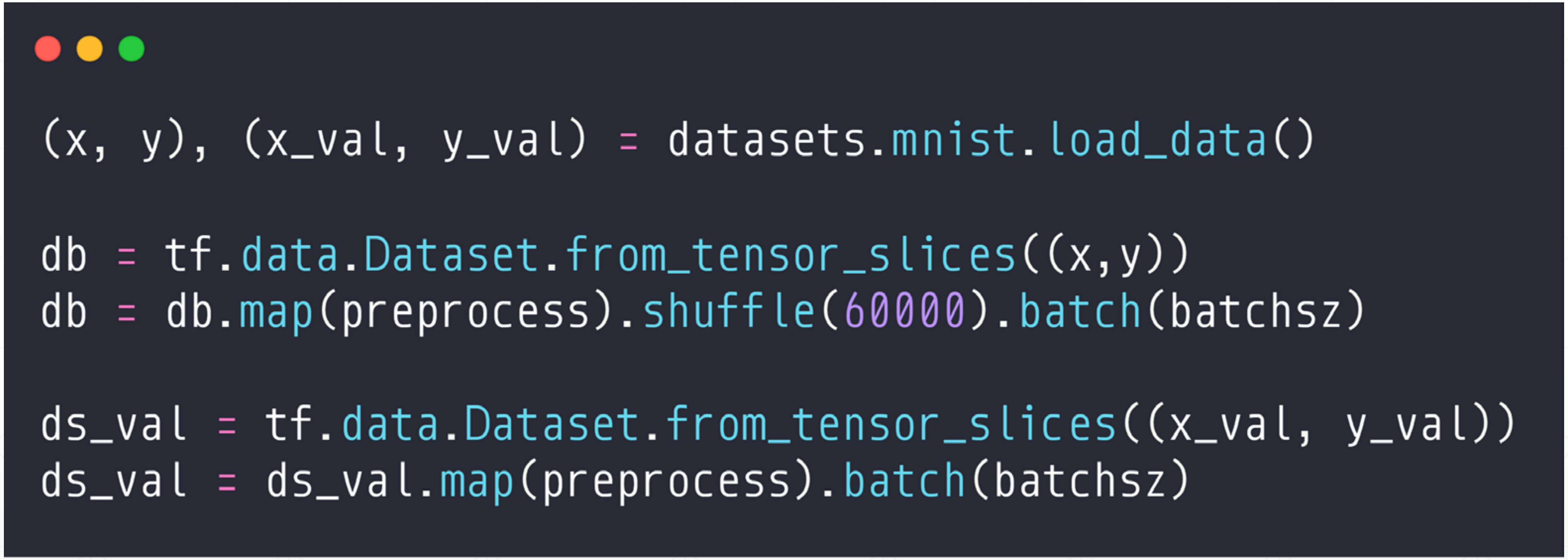
Splitting

Train Set



Test Set

Train, Test



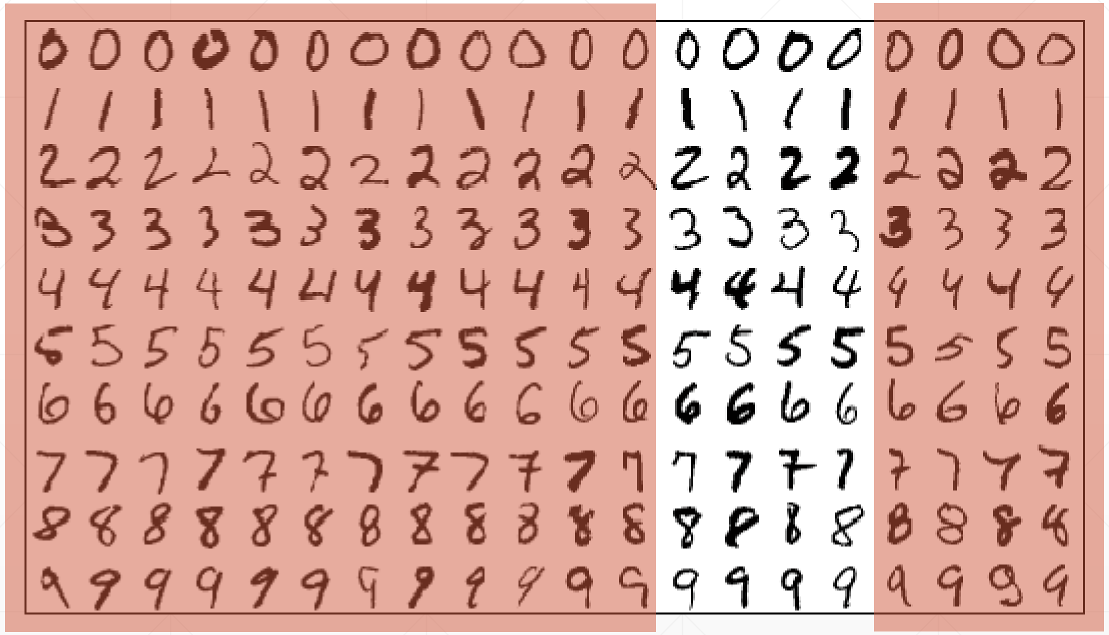
```
(x, y), (x_val, y_val) = datasets.mnist.load_data()

db = tf.data.Dataset.from_tensor_slices((x,y))
db = db.map(preprocess).shuffle(60000).batch(batchsz)

ds_val = tf.data.Dataset.from_tensor_slices((x_val, y_val))
ds_val = ds_val.map(preprocess).batch(batchsz)
```

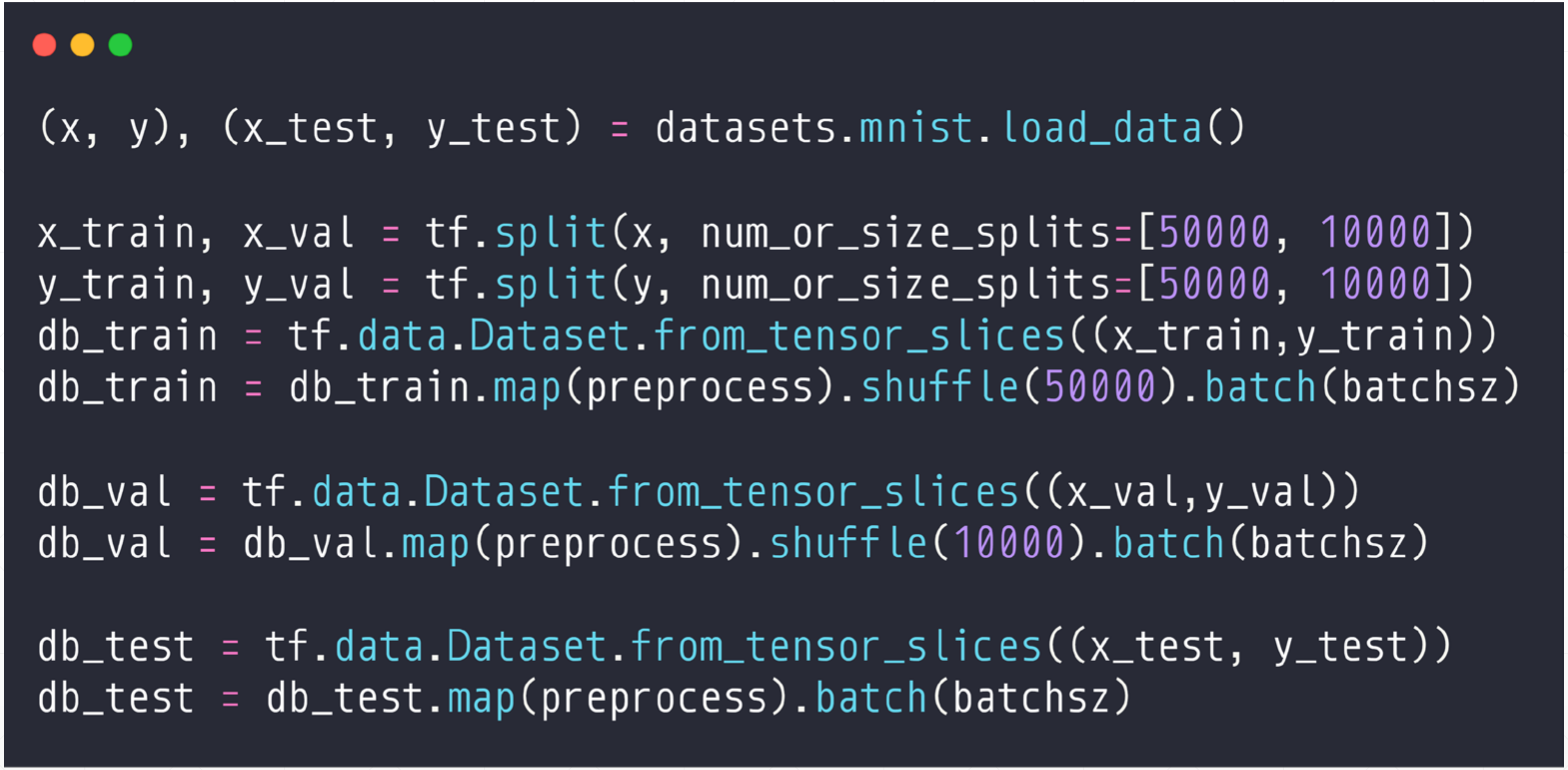
Splitting

Train Set



Test Set

Train, val, test



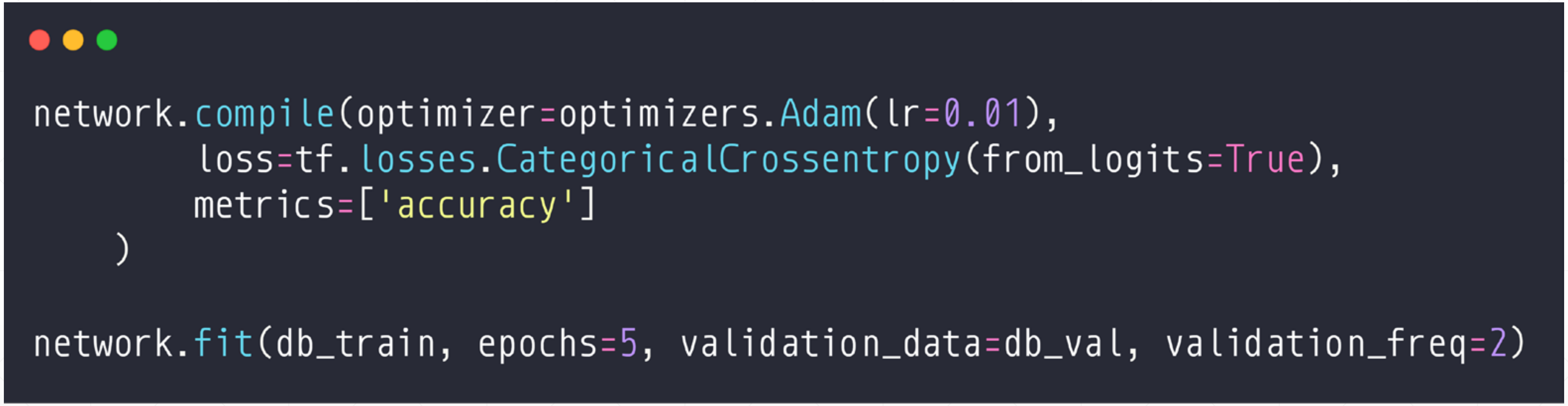
```
(x, y), (x_test, y_test) = datasets.mnist.load_data()

x_train, x_val = tf.split(x, num_or_size_splits=[50000, 10000])
y_train, y_val = tf.split(y, num_or_size_splits=[50000, 10000])
db_train = tf.data.Dataset.from_tensor_slices((x_train,y_train))
db_train = db_train.map(preprocess).shuffle(50000).batch(batchsz)

db_val = tf.data.Dataset.from_tensor_slices((x_val,y_val))
db_val = db_val.map(preprocess).shuffle(10000).batch(batchsz)

db_test = tf.data.Dataset.from_tensor_slices((x_test, y_test))
db_test = db_test.map(preprocess).batch(batchsz)
```

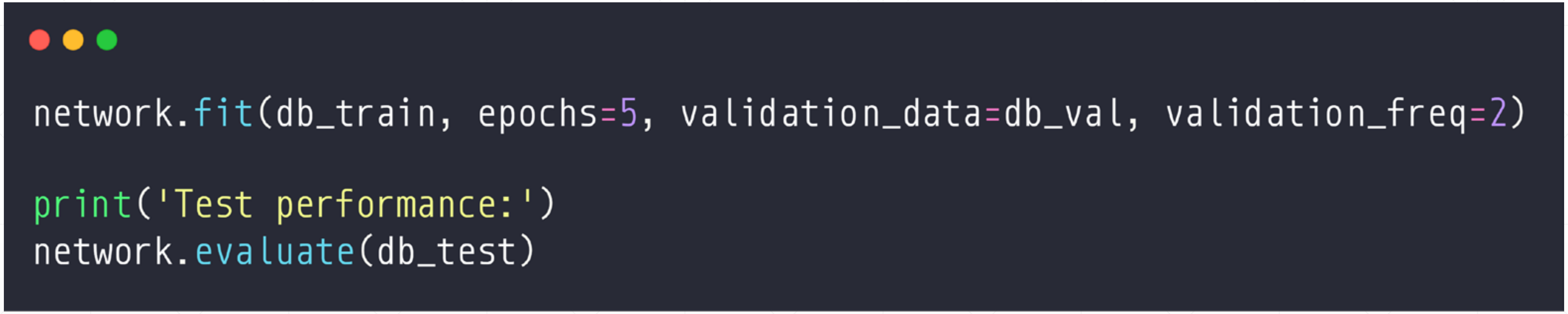
Evaluate during train



```
network.compile(optimizer=optimizers.Adam(lr=0.01),
        loss=tf.losses.CategoricalCrossentropy(from_logits=True),
        metrics=['accuracy']
    )

network.fit(db_train, epochs=5, validation_data=db_val, validation_freq=2)
```

Test after training

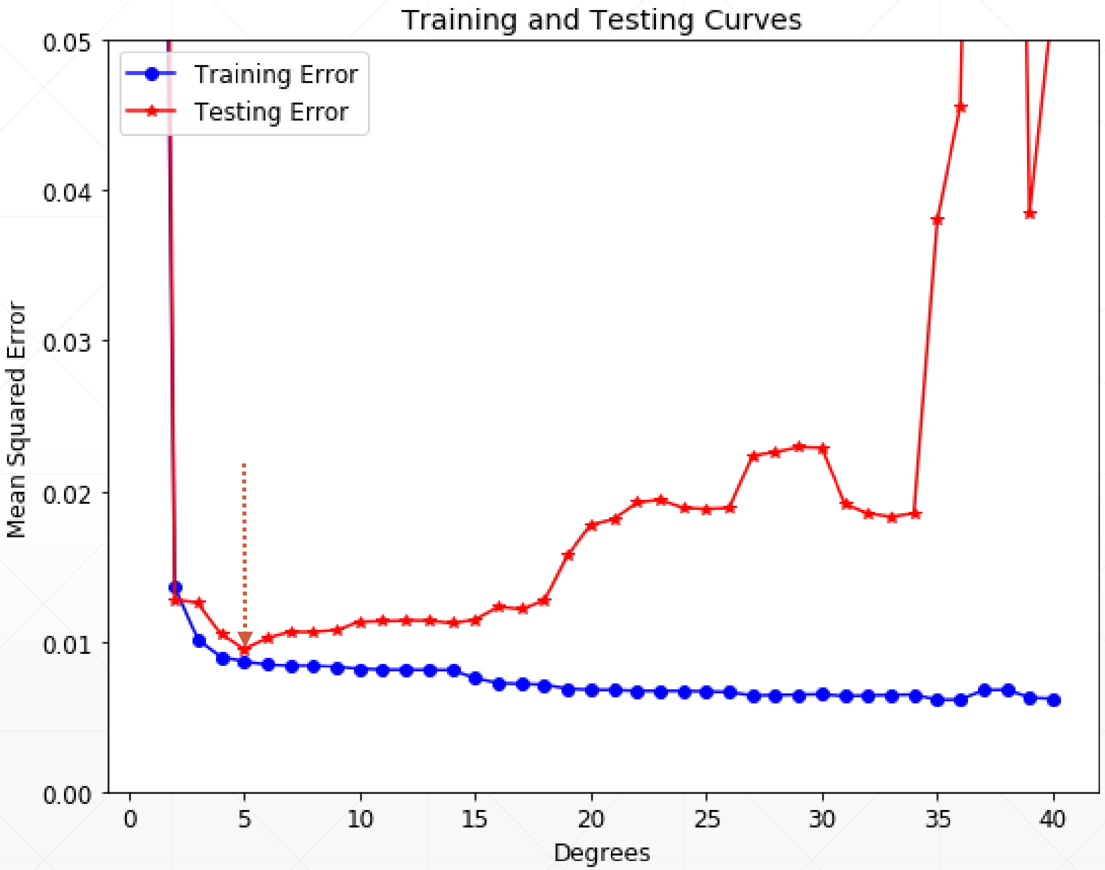


```
network.fit(db_train, epochs=5, validation_data=db_val, validation_freq=2)

print('Test performance:')
network.evaluate(db_test)
```

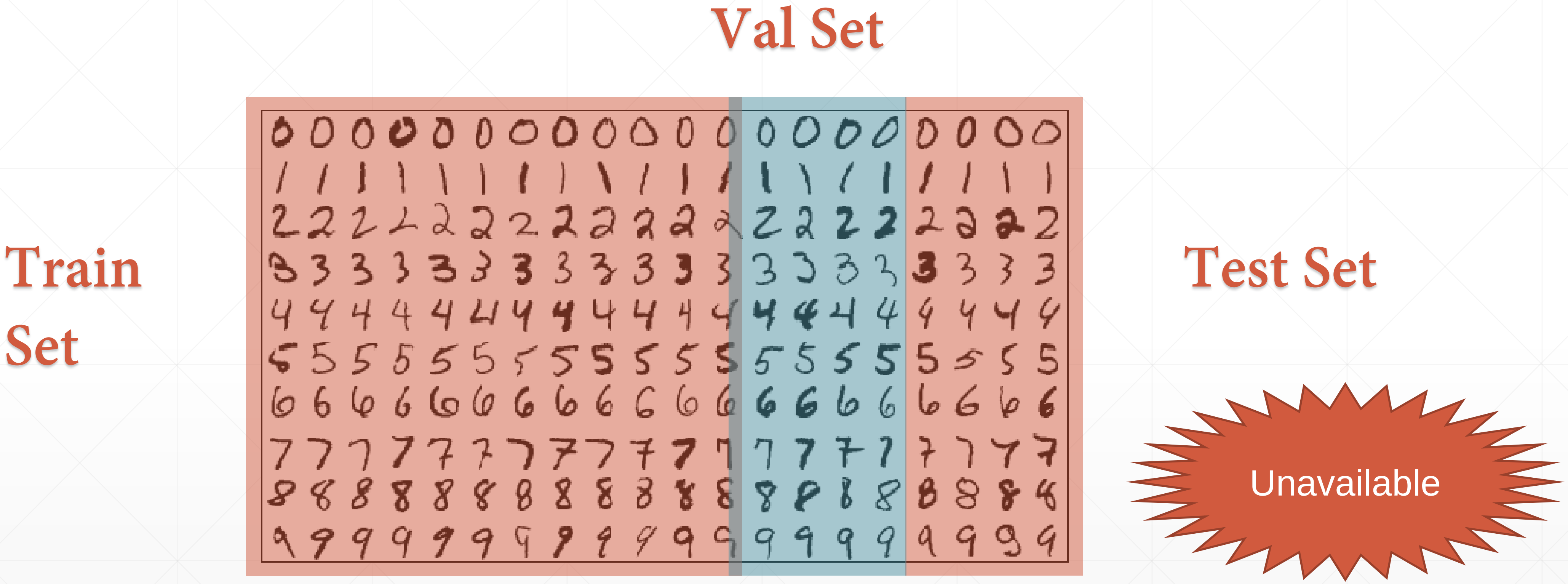
train test trade-off

Overfitt
ing



For others judge

- Kaggle



Val Set

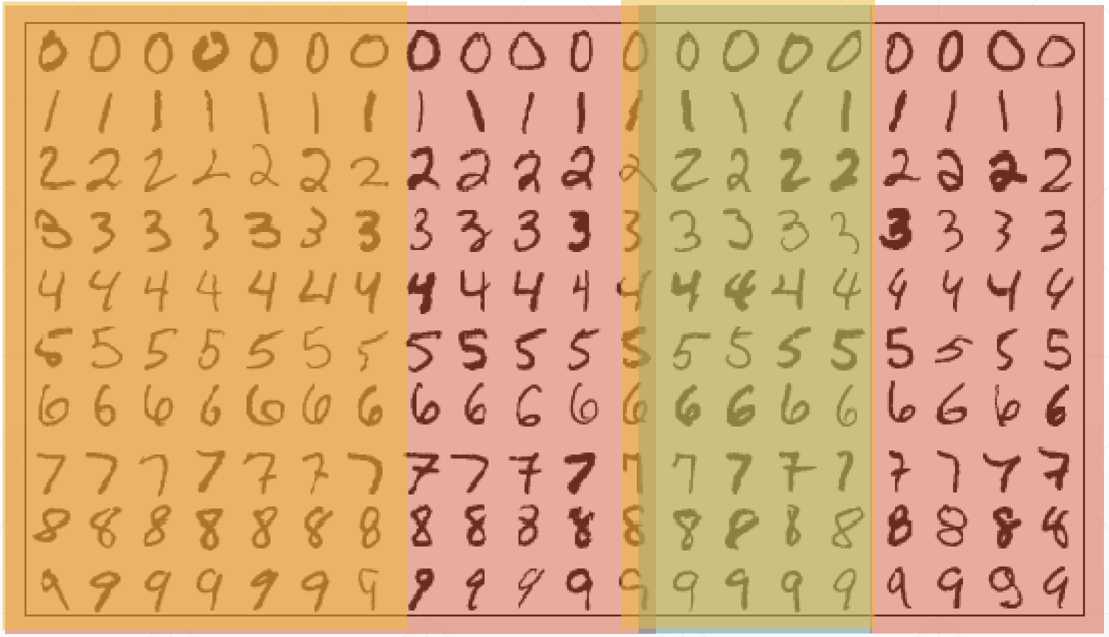
Train Set

Test Set

Unavailable

https://www.kaggle.com/

K-fold cross-validation



Val Set

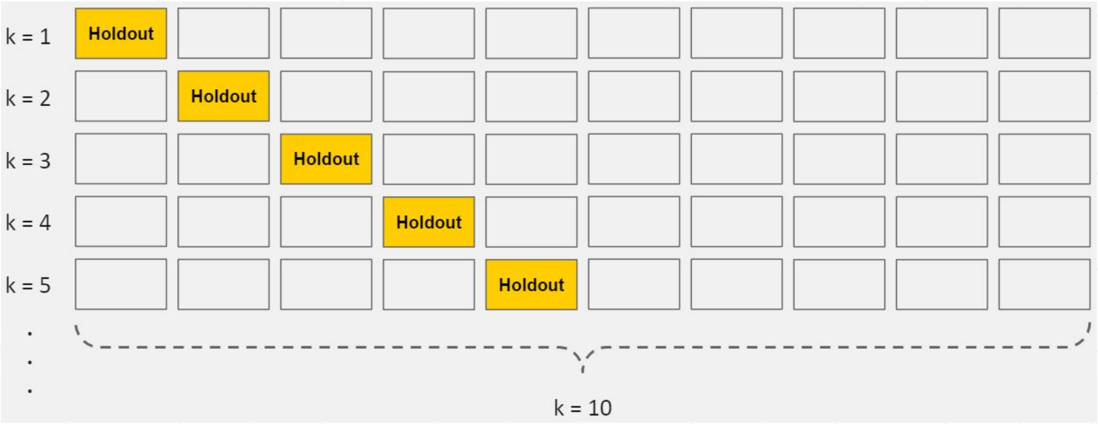
Train Set

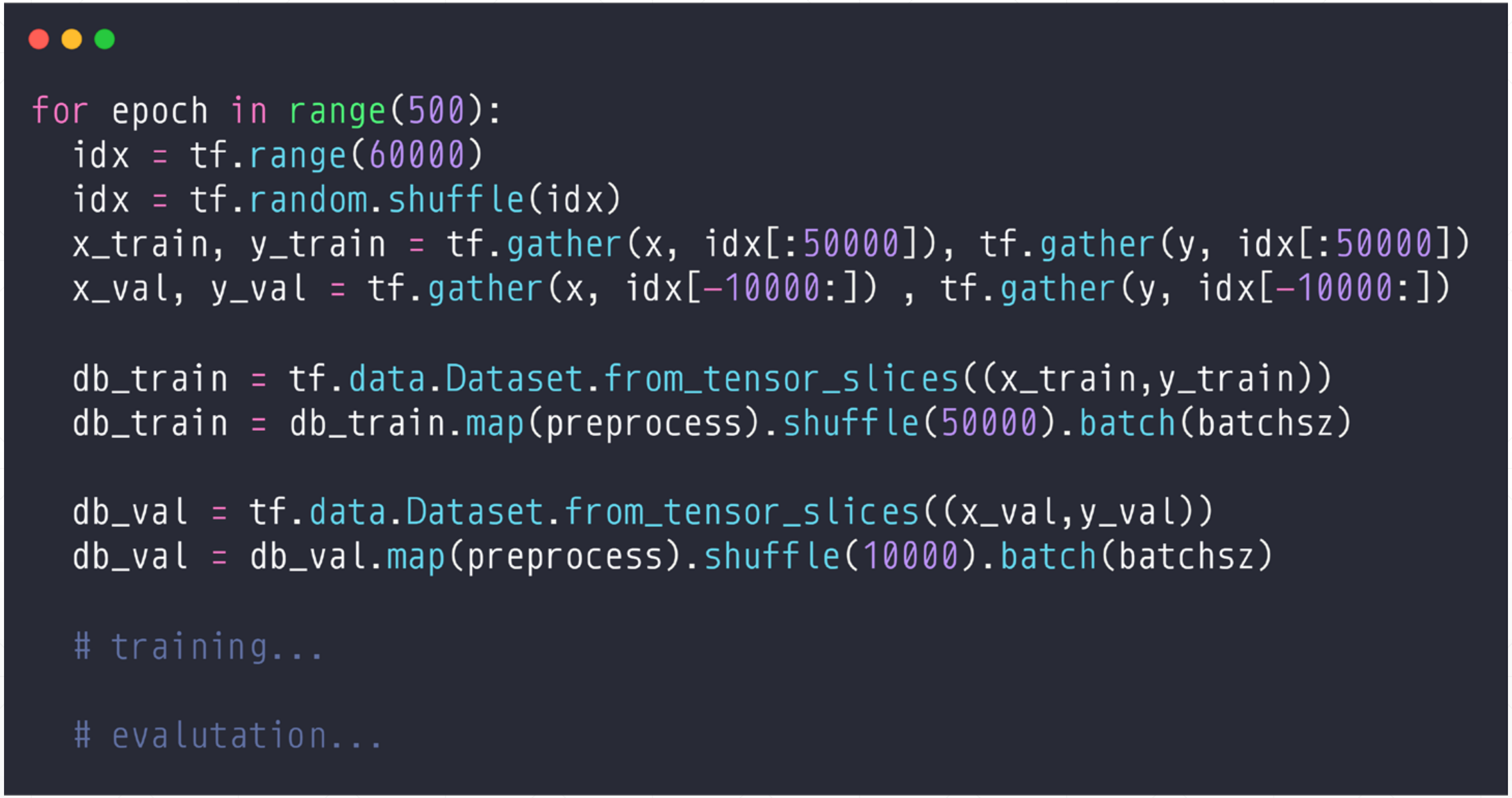
Test Set

k-fold cross validation

- merge train/val sets

- randomly sample 1/k as val set





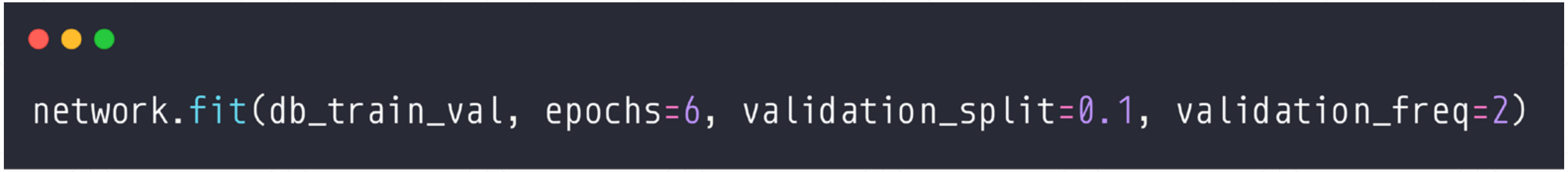
```
for epoch in range(500):
  idx = tf.range(60000)
  idx = tf.random.shuffle(idx)
  x_train, y_train = tf.gather(x, idx[:50000]), tf.gather(y, idx[:50000])
  x_val, y_val = tf.gather(x, idx[-10000:]) , tf.gather(y, idx[-10000:])

  db_train = tf.data.Dataset.from_tensor_slices((x_train,y_train))
  db_train = db_train.map(preprocess).shuffle(50000).batch(batchsz)

  db_val = tf.data.Dataset.from_tensor_slices((x_val,y_val))
  db_val = db_val.map(preprocess).shuffle(10000).batch(batchsz)

  # training...

  # evalutation...
```



```
network.fit(db_train_val, epochs=6, validation_split=0.1, validation_freq=2)
```

下一课时

Regularization

Thank You.