Flooding
PRs reviewed

Anubhav

PR #1

Adaptive Flooding Method

```
17 + # adaptive loss function
18 + def adaptive_flood_catergorical_crossentropy(b):
19 +     value = K.eval(b)
20 +     def crossentropy_loss(y_true, y_pred):
21 +         loss = tf.keras.losses.categorical_crossentropy(y_true, y_pred)
22 +         loss = tf.math.abs(loss - value) + value
23 +         return loss
24 +     return crossentropy_loss
25 +
26 + # adaptive rule
27 + def adapt(epoch):
28 +     if epoch > 0 and epoch % 2 == 0:
29 +         value = K.eval(b) - lamb
30 +         K.set_value(b, value)
31 +         print("Assigned new flooding value: %4f at epoch %d" % (value, epoch))
32 +
33 + # callback for adaptive loss function dependent on epoch
34 + adaptive_cb = LambdaCallback(on_epoch_end=lambda epoch, log: adapt(epoch))
```

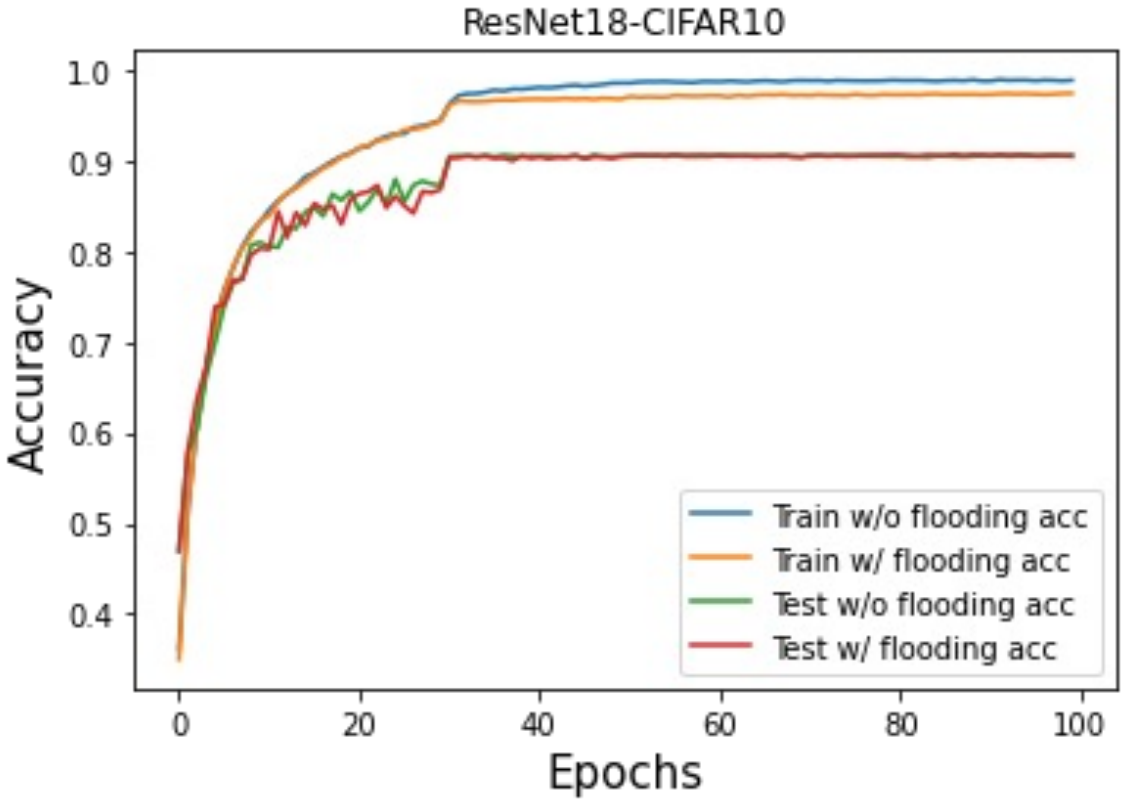
Code

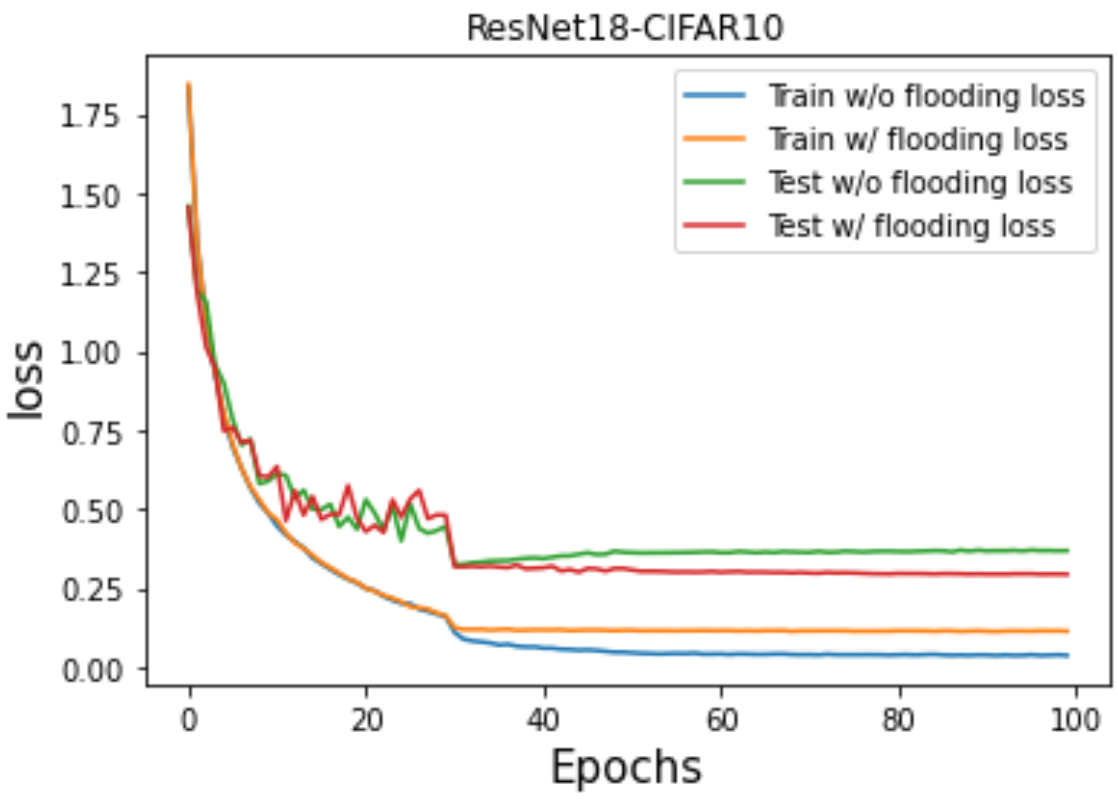
```
Shape of x_train: (60000, 28, 28) y_train: (60000,)
Shape of x_test: (10000, 28, 28) y_test: (10000,)
Training matrix shape (60000, 784)
Testing matrix shape (10000, 784)
Assigned new flooding value: 0.009900 at epoch 2
Assigned new flooding value: 0.009800 at epoch 4
Assigned new flooding value: 0.009700 at epoch 6
Assigned new flooding value: 0.009600 at epoch 8
313/313 - 2s - loss: 0.0726 - mse: 2.4370e-05 - acc: 0.9848 - 2s/epoch - 7ms/step
```

Results

PR #2

ResNet implementation for CIFAR10





PR #3

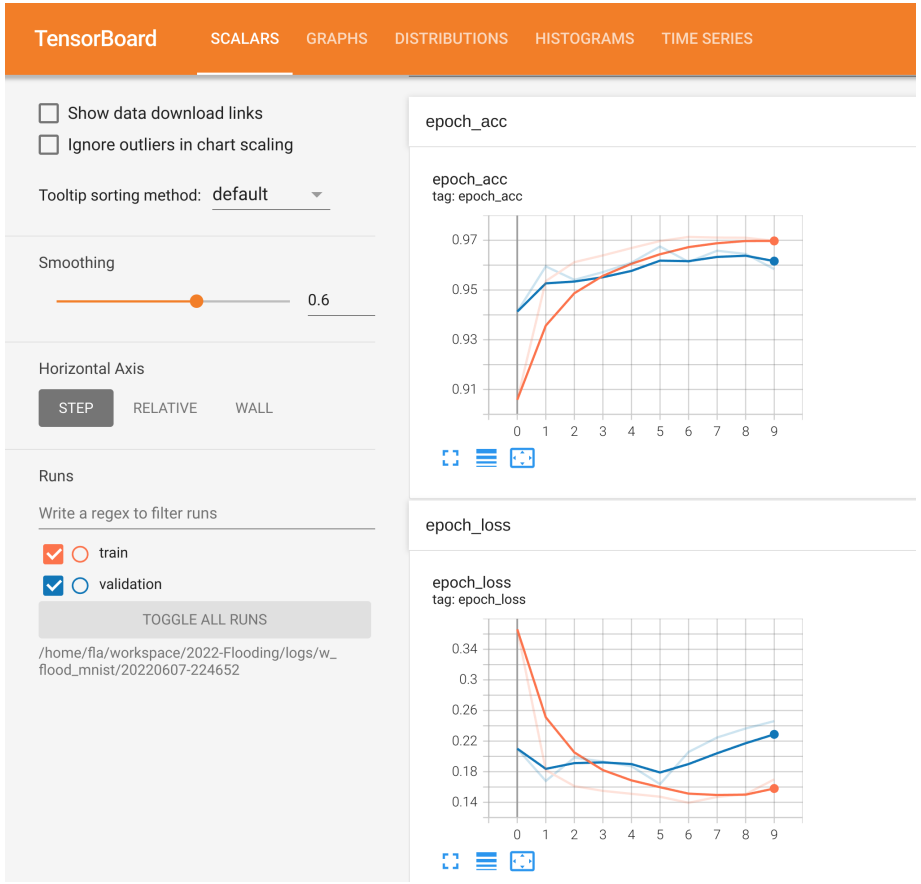
Tensorboard visualizer

16 models/mlp_mnist.py — Viewed

```
@@ -1,8 +1,10 @@
+ from gc import callbacks
  import numpy as np
  import tensorflow as tf
  from keras.layers import Dense
-
  import matplotlib.pyplot as plt
+ import datetime
+ import os

+ #add support for tensorboard
+ log_dir="logs/w_flood_mnist/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
+ tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1)
+
  SGD = tf.keras.optimizers.SGD(learning_rate=0.1, momentum=0.9,)
  model.compile(loss=flood_categorical_crossentropy, optimizer=SGD, metrics=["mse", "acc"])
- history = model.fit(x_train, y_train, epochs=100, validation_data=(x_test, y_test))
+ history = model.fit(x_train, y_train, epochs=100, validation_data=(x_test, y_test), callbacks=[tensorboard_callback])
  model.evaluate(x_test,  y_test, verbose=2)

+ #add support for tensorboard
+ log_dir_1="logs/wo_flood_mnist/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
+ tensorboard_callback_1 = tf.keras.callbacks.TensorBoard(log_dir=log_dir_1, histogram_freq=1)
+
  SGD = tf.keras.optimizers.SGD(learning_rate=0.1, momentum=0.9,)
  model1.compile(loss="categorical_crossentropy", optimizer=SGD, metrics=["mse", "acc"])  #using categorical_crossentropy loss
- history1 = model1.fit(x_train, y_train, epochs=100, validation_data=(x_test, y_test))
+ history1 = model1.fit(x_train, y_train, epochs=100, validation_data=(x_test, y_test),callbacks=[tensorboard_callback_1])
  model1.evaluate(x_test,  y_test, verbose=2)
```



PR #3

Update .gitignore