Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

DECLARATION

I understand that this is an individual assessment and that collaboration is not permitted. I have not received any assistance with my work for this assessment. Where I have used the published work of others, I have indicated this with appropriate citation.

I have not and will not share any part of my work on this assessment, directly or indirectly, with any other student.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write."

I understand that by returning this declaration with my work, I am agreeing with the above statement. ☑

Name: Xiangyu Zheng

Date: 27/4/2022

CS7CS2 Optimisation for Machine Learning final assignment

Xiangyu Zheng 21331025

1 experiment design

This experiment is to compare the performance of adam and constant step in SGD on different datasets and different models. To find differences between optimizers, this experiment uses three standard datasets in keras: MNIST, CIFAR, and Imdb. Similarly, two models, neural network and logistic regression, were also chosen to evaluate the performance of adam and SGD constant on different datasets. Next, hyperparameters of adam and constant step are selected by a population based search algorithm and compared with the default parameters to find the best parameters for inter-algorithm comparison. The performance of the optimizer is evaluated by the loss value and accuracy in multiple training and test and training sets.

2 Methodology

2.1 Dataset

- MNIST - is a dataset of 10 digital images. Each image is 28*28 pixels, for a total of one channel. Each pixel point of the image is converted to between 0 and 1 before training.
- CIFAR10 - is a dataset consisting of images of 10 different objects such as birds, deer and dogs. Each image is 32*32 pixels and has a total of 3 channels. The value of each pixel point of the image needs to be converted to between 0 and 1 before training.
- Imdb - is a binary dataset with hundreds of terms used to judge whether a movie is rated positive (1) or negative (0)

2.2 Model

- neural network

  1. MNIST

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d (Conv2D)              (None, 13, 13, 32)        320

 conv2d_1 (Conv2D)            (None, 6, 6, 64)          18496

 flatten (Flatten)            (None, 2304)              0

 dense (Dense)                (None, 10)                23050

=================================================================
Total params: 41,866
Trainable params: 41,866
Non-trainable params: 0
```

In the model for the MNIST dataset, The input layer is 32 filters, the kernel is 3*3, and the sliding step size is (2, 2). The activation function is relu.
The second layer is 64 filters, and then the flatten operation is performed.
The output layer has 10 output units corresponding to 10 categories, the activation function is softmax, and L2 regularization is used.

2. CIFAR10

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d (Conv2D)              (None, 32, 32, 16)        448

 conv2d_1 (Conv2D)            (None, 16, 16, 16)        2320

 conv2d_2 (Conv2D)            (None, 16, 16, 32)        4640

 conv2d_3 (Conv2D)            (None, 8, 8, 32)          9248

 dropout (Dropout)            (None, 8, 8, 32)          0

 flatten (Flatten)            (None, 2048)              0

 dense (Dense)                (None, 10)                20490

=================================================================
Total params: 37,146
Trainable params: 37,146
Non-trainable params: 0
```

The input layer of the CNN model of CIFAR10 is a Conv2D with 16 filters, a 3*3 kernel, the edges are filled with 0, and the activation function is relu.
The second layer also uses 16 filters, the sliding step is (2, 2), the edge is filled with 0, and the activation function is relu.
The third and fourth layers are both increased to 32 filters, the others are the same as the first and second layers.
Then the model does dropout and flatten processing.
The output layer is also 10 units, the activation function is softmax, and L1 regularization is used.

3. Imdb

```
train_data = keras.preprocessing.sequence.pad_sequences(train_data,value=
                                          word_index["<PAD>"],padding='post'
                                          ,maxlen=256)
```

First convert the training set and test set into a matrix of data length*maxlen

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 embedding (Embedding)        (None, None, 16)          160000

 global_average_pooling1d (G  (None, 16)                0
 lobalAveragePooling1D)

 dense (Dense)                (None, 16)                272

 dense_1 (Dense)              (None, 1)                 17

=================================================================
Total params: 160,289
Trainable params: 160,289
Non-trainable params: 0
```

The first is the embedding layer, which is an integer-encoded vocabulary, which is trained on the input data to obtain an embedding vector and output to the next layer.[1].
The second layer calculates the average dimension of the input sequence in the simplest way, and outputs.
The third layer is a fully connected layer with 16 filters.
The output layer uses the sigmoid activation function to output a one-dimensional 0-1 number representing the probability.

- logistic regression
  Since logistic regression has good convex optimization performance, there is no need to worry about the local minimum problem, and it is suitable for comparing different optimizers, so the logistic regression model is used in this experiment[2].
  keras is used to implement logistic regression[3], the binary_crossentropy loss function can be used in the binary classification, and the categorical_crossentropy is used in the multi-classification task[4].

  1. MNIST and CIFAR10
     Since both MNIST and CIFAR10 datasets are divided into 10 categories, the models are the same.

```
model.add(Dense(10, activation='softmax', input_dim=x_train.shape[1]))
model.compile(loss="categorical_crossentropy", optimizer=optimizer, metrics=[
                                                "accuracy"])
```

     The output units are all 10, and the activation function is softmax,Just the input dimension is different.
     for CIFAR10, Convert to (data_length, 3072)(32*32*3):

```
x_train = x_train.reshape((-1, 3072))
x_test = x_test.reshape((-1, 3072))
```

     for MNIST, Convert to (data_length, 784)(28*28):

```
x_train = x_train.reshape((-1, 784))
x_test = x_test.reshape((-1, 784))
```

  2. Imbd

```
model.add(Dense(1, activation='sigmoid', input_dim=train_data.shape[1]))
model.compile(loss="binary_crossentropy", optimizer=optimizer, metrics=["
                                                accuracy"])
```

     There is only one output layer unit, the activation function is sigmoid, and the loss function is binary_crossentropy

2.3 Optimizer

- Adam
  adam is roughly equivalent to RMSprop + heavy ball
  First use exponentially weighted average for gradient, square of gradient:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

  Then bias-correct it:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

  The update formula of the final weight is as follows:

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_{t-1}}{\sqrt{\hat{v}_{t-1}} + \epsilon}$$

  The main parameters are: α,$\beta 1$ and $\beta 2$. The adam optimizer can be used directly in keras:

```
optimizer = keras.optimizers.Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999)
```

- SGD constant step
  SGD constant step is a basic optimizer,the main parameters is α.In keras it is possible to use the SGD optimizer without nesterov and with momentum=0.0[5]:

```
optimizer = keras.optimizers.SGD(learning_rate=0.01, momentum=0.0, nesterov=False)
```

2.4 select parameters

Population Based Search method is used to find the best hyperparameters[6].First, randomly select N samples of adam or logistic regression hyperparameters, and choose the best ones. Then start iterating, randomly select samples around the best few samples, and replace if they perform better than them.

2.5 performance evaluation method

In the MNIST and CIFAR10 datasets, the loss value and prediction accuracy of the categorical crossentropy of the training set and test set are used to measure the performance of the model.
In Imdb data, the loss value of binary crossentropy is replaced to measure the performance of the model.

3 Evaluation

- cifar10

  First, the Population Based Search algorithm is used to find the best parameters of adam, but the model does not perform well compared to the default parameters. The default parameters of adam are used, which are alpha=0.001, beta1=0.9 and beta2 =0.999. The optimal batch size is 128. The SGD constant step optimizer can use the parameter search algorithm to obtain the optimal alpha of 0.05. The batch size is 128 for comparison, and run The two models get Figures 1, 2, 3, and 4.
  We can see from Figure 1 that under the cifar10 and cnn models, the performance of adam is better than that of SGD constant step, and the iteration speed is also faster.
  Figure 2 reflects the performance of Adam and SGD constant step on the logistic regression model and the cifar10 dataset, both on the training set and the test set. But still adam is better.
  As can be seen from Figure 3 and Figure 4, both Adam and SGD have a certain degree of randomness, and each run is slightly different.

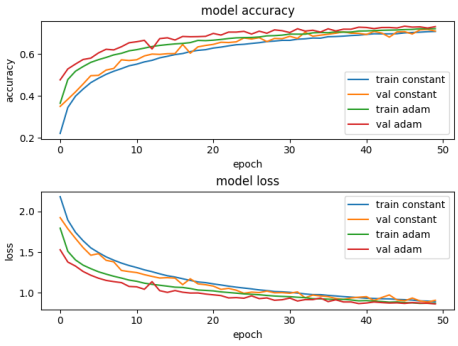

Figure 1: comparison of adam and constant step under cifar10 and cnn model

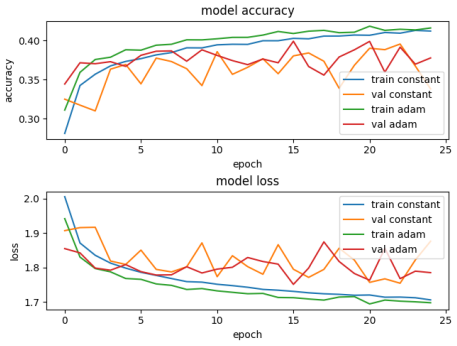

Figure 2: comparison of adam and constant step under cifar10 and RL model

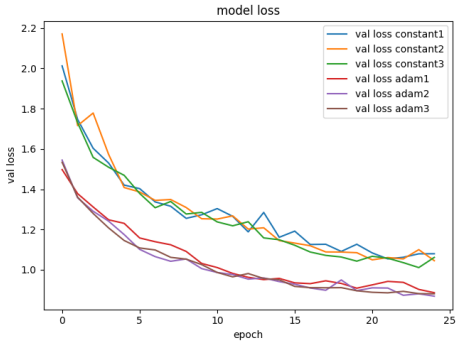

Figure 3: adam and constant step run multiple times under cnn and cifar10

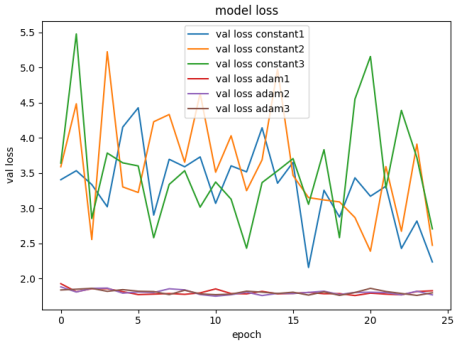

Figure 4: adam and constant step run multiple times under RL and cifar10

- MNIST
  In the MNIST dataset, the default hyperparameters in keras are used, namely Adam: learning rate = 0.001, beta1=0.9 and beta2 =0.99. The learning rate of SGD constant step = 0.01. Figures 5 and 6 reflect that adam outperforms SGD on either CNN or logistic regression models. But adam shows a little bit of overfitting when adding dropout and L1 regularization. Figures 7 and 8 reflect that the random algorithm is still slightly different from run to run

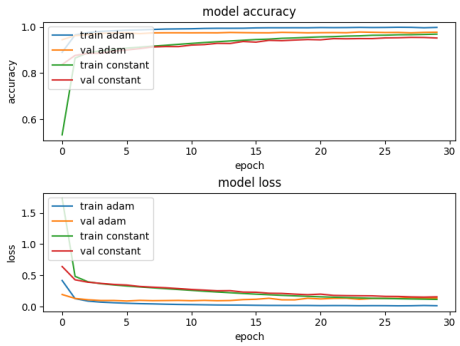

Figure 5: comparison of adam and constant step under MNIST and cnn model

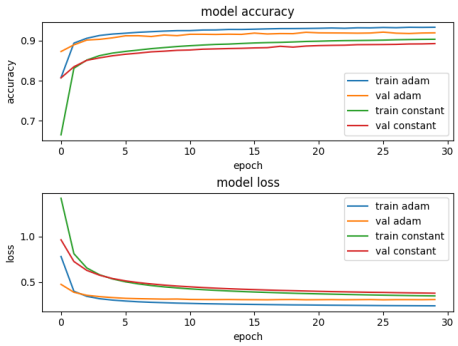

Figure 6: comparison of adam and constant step under MNIST and RL model

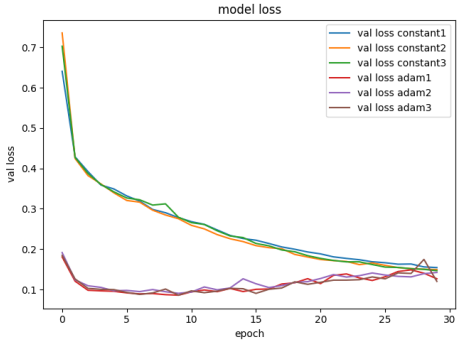

Figure 7: adam and constant step run multiple times under cnn and MNIST

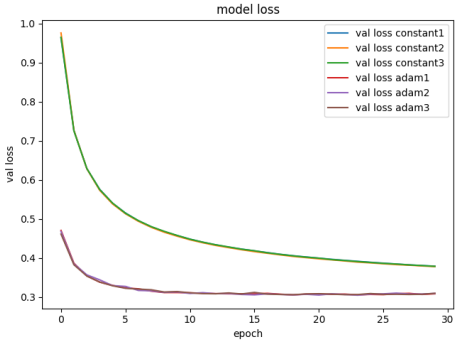

Figure 8: adam and constant step run multiple times under RL and MNIST

- Imdb
  In the non-image dataset Imdb, Adam still uses the default hyperparameters. While SGD selects learning rate=0.1 under the cnn model with the help of the parameter search algorithm, and

selects learning rate=0.3 under the logistic regression model.
We can see from Figure 9 and Figure 10 that adam is much better than SGD in terms of accuracy and loss value, but there is a serious overfitting phenomenon.In the later period, there was also a situation that could not be converged. This is due to the fact that the later adam learning step size is too small. SGD has a slow iteration.

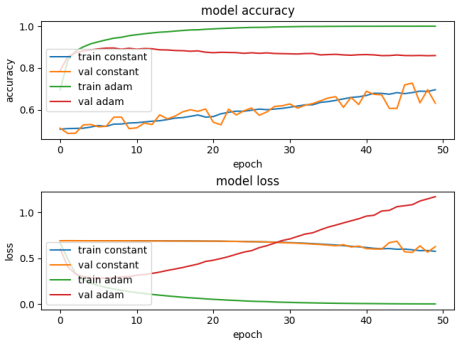

Figure 9: comparison of adam and constant step under Imdb and cnn model

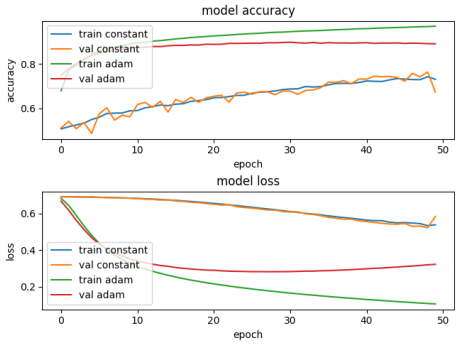

Figure 10: comparison of adam and constant step under Imdb and RL model

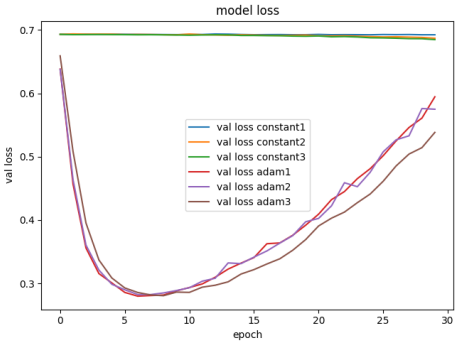

Figure 11: adam and constant step run multiple times under cnn and Imdb

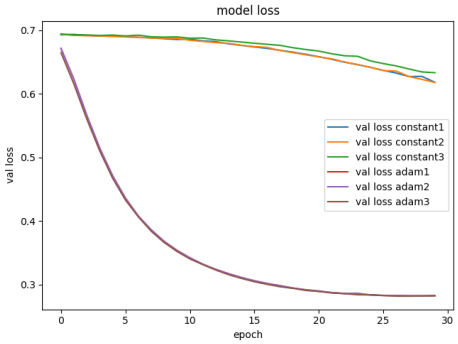

Figure 12: adam and constant step run multiple times under RL and Imdb

References

[1] https://tensorflow.google.cn/tutorials/keras/text_classification

[2] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[3] https://www.marktechpost.com/2021/04/08/logistic-regression-with-keras/

[4] Ren, Y., Zhao, P., Sheng, Y., Yao, D. and Xu, Z., 2017, August. Robust softmax regression for multi-class classification with self-paced learning. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (pp. 2641-2647).

[5] https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/SGD

[6] Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K. and Fernando, C., 2017. Population based training of neural networks. arXiv preprint arXiv:1711.09846.