Experimentation of CNN with Fashion MNIST Dataset

Prequisites

- Tensorflow 1.12
- Keras
- Python 3.x

Dataset

Fashion-MNIST is dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. Zalando intends Fashion-MNIST to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits.

How to Use

1. Open this repository notebook
2. Run each step by pressing "play" button or CTRL + ENTER
3. Change the parameters in "Parameters Section"

Run the Models

1. Go to section running models
2. Select model that want to evaluate
3. Run evaluate

Training Steps

Importing required modules

```
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import StratifiedShuffleSplit
from keras import backend as k

import numpy as np
import matplotlib.pyplot as plt
```

Load Datasets

```
(imageTrain, labelTrain), (imageTest, labelTest) = tf.keras.datasets.fashion_mnist.load_data
```

Make array of labels and split training data into validation data

```
labels = {0 : "T-shirt/top", 1: "Trouser", 2: "Pullover", 3: "Dress", 4: "Coat", 5: "Sandal'

imageTrain = np.expand_dims(imageTrain, -1)
imageTest = np.expand_dims(imageTest, -1)

sss = StratifiedShuffleSplit(n_splits=5, random_state=0, test_size=1/6)
trainIndex, validIndex = next(sss.split(imageTrain, labelTrain))
imageValid, labelValid = imageTrain[validIndex], labelTrain[validIndex]
imageTrain, labelTrain = imageTrain[trainIndex], labelTrain[trainIndex]
```

Perform Mean Subtraction and Normalization of Datasets (a)

- Scaling down range of input value into between 0 and 1 instead of 0 - 255

```
imageTrain = imageTrain / 255
imageValid = imageValid / 255
imageTest = imageTest / 255
```

- Mean Substraction of data

```
meanSubt = np.mean(imageTrain)

imageTrain = imageTrain - meanSubt
imageValid = imageValid - meanSubt
imageTest = imageTest - meanSubt
```

- Normalization

```
stdDev = np.std(imageTrain)

imageTrain = imageTrain / stdDev
imageValid = imageValid / stdDev
imageTest = imageTest / stdDev
```

Mean subtraction and normalization using image training data to make sure it's same across all datasets that we use. This process necessary because we want to make our data sparse before we train.

Initializer

We experiment with two weight initialization: He initialization and Xavier Initialization

We can see the result in below table:

	Xavier Initializer	He Initializer
Accuracy	0.9663	0.9675
Loss	0.0891	0.0863

	Xavier Initializer	He Initializer
Validation Accuracy	0.9194	0.9184
Validation Loss	0.284	0.2929
Test Accuracy	0.9138	0.9159
Test Loss	0.3143	0.3164