TP2 IA

Yining BAO 3700515

Partie 1

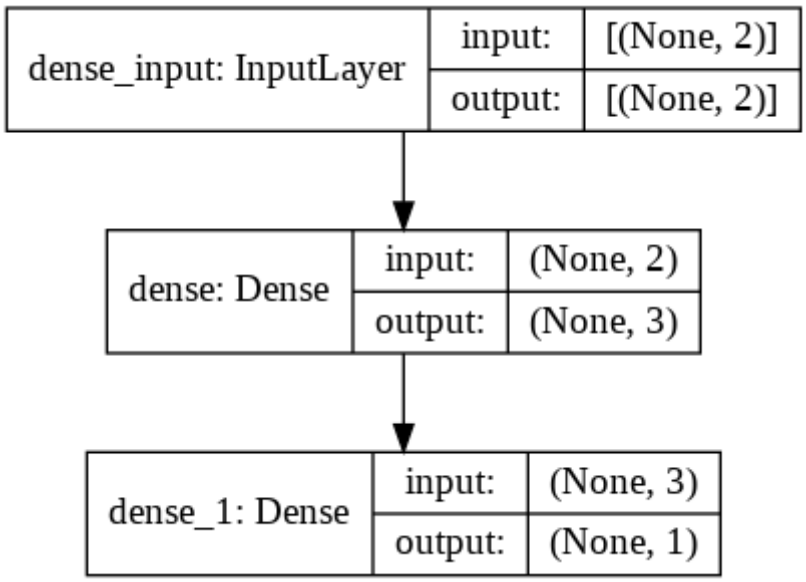
1.1.1

On a bien obtenu les dimensions de ces arrays

```
C:\Users\Kzel>set PYTHONIOENCODING=utf8 && "D:\Program\
2021-04-16 08:56:36.292432: I tensorflow/stream_executo
(4, 2)
(4, 1)
```

1.1.2

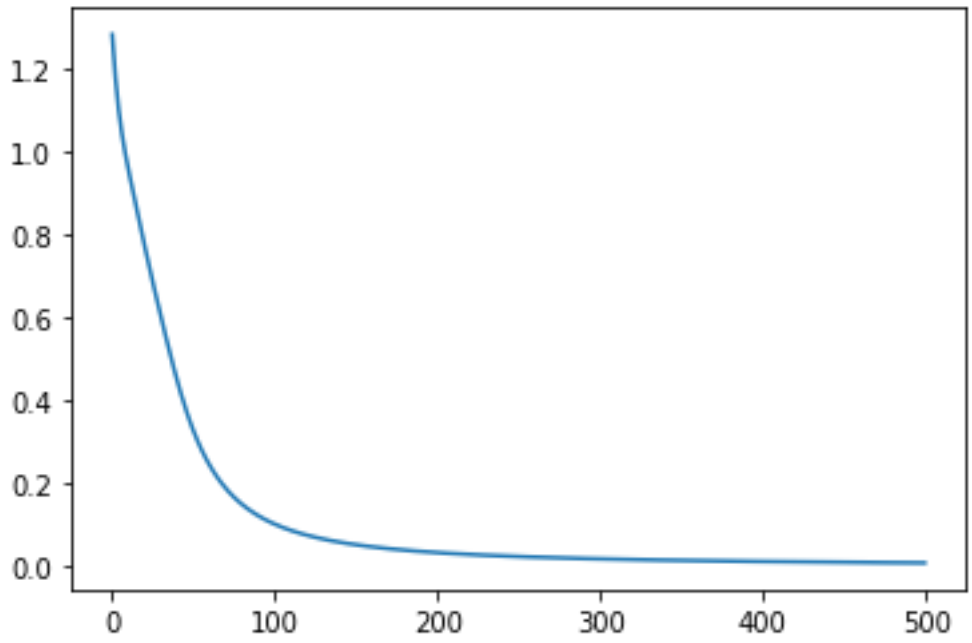
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 3)                 9
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 4
=================================================================
Total params: 13
Trainable params: 13
Non-trainable params: 0
_________________________________________________________________
```



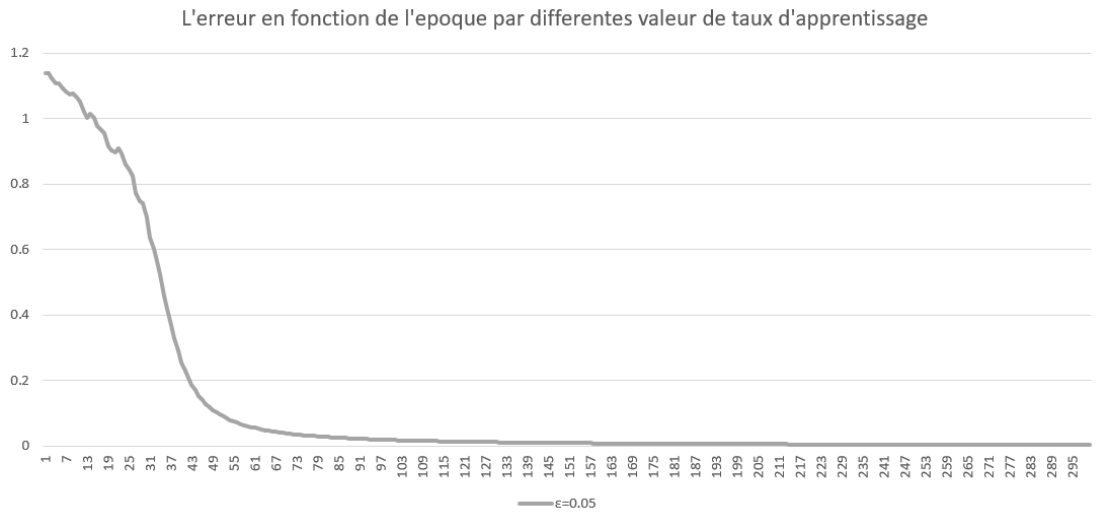
Pour Param 9, parce-que on a 2 entrées 2+1BIAS 3*3neurones, on a obtenu 9

Pour Param 4, 1sortie*3neurones+1sortie=4

1.2



Le graphique qu'on a obtenu en TP1



On a observé l'erreur baisse plus vite dans TP1. Elle est atteinte a 0 avec moins d'époque en TP1

Partie2

2.1.1

Après la normalisation

On a obtenu la moyenne de train_xdata_norm est 0 et l'écart-type est 1

```
110.dll
[-1.01541438e-16  1.09923072e-17  1.80933376e-15 -7.80453809e-17
 -5.25047552e-15  6.43187374e-15  2.98441140e-16  4.94653823e-16
  1.12671149e-17 -1.05526149e-16  2.36614908e-14  5.96710525e-15
  6.13920356e-16]
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

2.1.2

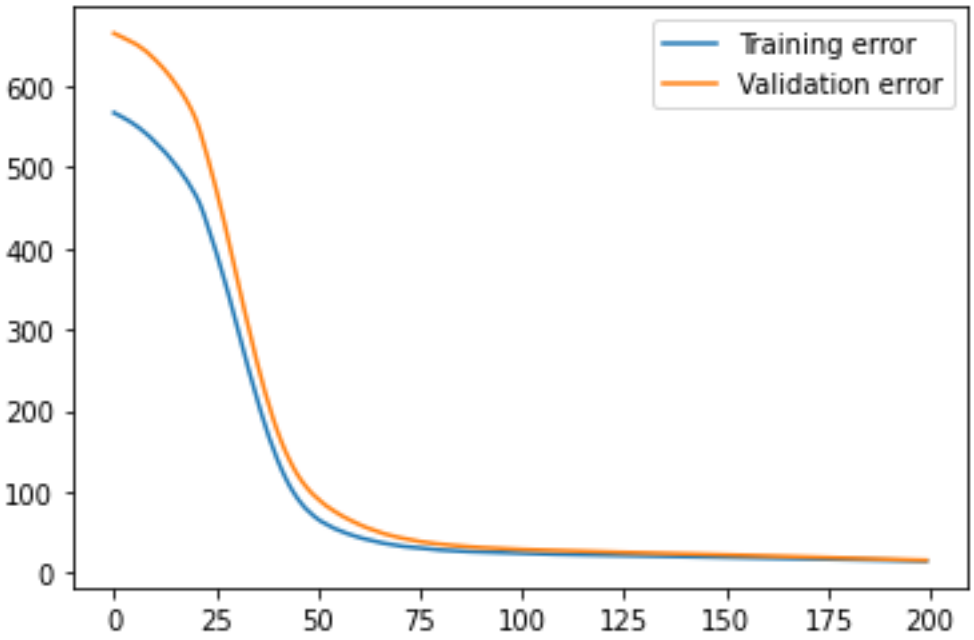
D'après m.summary() , on a bien obtenu les params

Param 280 = 20 neurones *(13 entrees+1 BIAS)

Param 21 = 20 neurones *1 sortie + 1 sortie

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 20)                280
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 21
=================================================================
Total params: 301
Trainable params: 301
Non-trainable params: 0
```

2.3



Non, on n'observe pas le surapprentissage. Lorsque l'erreur d'apprentissage (Training error) baisse, l'erreur de validation(Validation error) baisse aussi

```
plt.legend()

test_xdata_norm = test_xdata
testx_moy = np.mean(test_xdata,axis=0)
test_xdata_norm -= testx_moy
testx_std = np.std(test_xdata,axis=0)
test_xdata_norm /= testx_std
erreur =  m.evaluate(test_xdata_norm , test_ydata)
print("Erreur est:",np.sqrt(erreur))
```

```
PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL                    1: Code
Epoch 196/200
11/11 [==============================] - 0s 4ms/step - loss: 15.7302 - val_loss: 14.2070
Epoch 197/200
11/11 [==============================] - 0s 4ms/step - loss: 13.3907 - val_loss: 14.1692
Epoch 198/200
11/11 [==============================] - 0s 4ms/step - loss: 12.6195 - val_loss: 14.0652
Epoch 199/200
11/11 [==============================] - 0s 4ms/step - loss: 16.9482 - val_loss: 13.9809
Epoch 200/200
11/11 [==============================] - 0s 4ms/step - loss: 13.1725 - val_loss: 13.9568
4/4 [==============================] - 0s 1ms/step - loss: 19.3440
Erreur est: 4.398181643590853
```

D'après l'evaluate, on trouve l'erreur est 4,40 elle est proche aux valeurs du leaderborad Kaggle

Si on change l'activation par LeakyReLU α=0.2

```
23    # 2.1.2
24    m = Sequential()
25    m.add(Dense(20, input_shape=(13,)))
26    #m.add(Dense(20, activation="relu",nput_shape=(13,)))
27    m.add(LeakyReLU(alpha=0.2))
28    m.add(Dense(1))
29    #m.add(Dense(1, activation="relu"))
30    m.add(LeakyReLU(alpha=0.2))
31    m.summary()
32
33    # 2.1.3
34    m.compile(optimizer="adam", loss="mean_squared_error")
35    history = m.fit(train_xdata_norm, train_ydata, epochs=200 ,
```

PROBLÈMES SORTIE CONSOLE DE DÉBOGAGE TERMINAL

```
Epoch 196/200
11/11 [==============================] - 0s 4ms/step - loss: 17.7031
Epoch 197/200
11/11 [==============================] - 0s 4ms/step - loss: 18.9418
Epoch 198/200
11/11 [==============================] - 0s 4ms/step - loss: 21.7201
Epoch 199/200
11/11 [==============================] - 0s 4ms/step - loss: 16.7434
Epoch 200/200
11/11 [==============================] - 0s 4ms/step - loss: 18.2321
4/4 [==============================] - 0s 2ms/step - loss: 17.8425
Erreur est: 4.22404204574524
```

L'erreur baisse a 4,22

Si on change l'époque par 400

```
34  m.compile(optimizer="adam", loss="mean_squared_error")
35  history = m.fit(train_xdata_norm, train_ydata, epochs=400 ,validation_split=0.15)
36  plt.plot(history.epoch, history.history["loss"], label="Training error")
37  plt.plot(history.epoch, history.history["val_loss"], label="Validation error")
38  plt.legend()
39
40  test_xdata_norm = test_xdata
41  testx_moy = np.mean(test_xdata,axis=0)
42  test_xdata_norm -= testx_moy
43  testx_std = np.std(test_xdata,axis=0)
44  test_xdata_norm /= testx_std
45  erreur =  m.evaluate(test_xdata_norm , test_ydata)
46  print("Erreur est:",np.sqrt(erreur))
```

```
PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL                1: Code
Epoch 396/400
11/11 [==============================] - 0s 4ms/step - loss: 10.8231 - val_loss: 17.8408
Epoch 397/400
11/11 [==============================] - 0s 4ms/step - loss: 11.8930 - val_loss: 17.7917
Epoch 398/400
11/11 [==============================] - 0s 4ms/step - loss: 14.9461 - val_loss: 17.7543
Epoch 399/400
11/11 [==============================] - 0s 4ms/step - loss: 14.9946 - val_loss: 17.7157
Epoch 400/400
11/11 [==============================] - 0s 5ms/step - loss: 11.9340 - val_loss: 17.7428
4/4 [==============================] - 0s 1ms/step - loss: 16.9604
Erreur est: 4.1182982594998165
```

L'erreur baisse à 4,12

Partie 3

3.1.2

```
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 784)               615440
_________________________________________________________________
dense_1 (Dense)              (None, 100)               78500
_________________________________________________________________
dense_2 (Dense)              (None, 10)                1010
=================================================================
Total params: 694,950
Trainable params: 694,950
Non-trainable params: 0
```

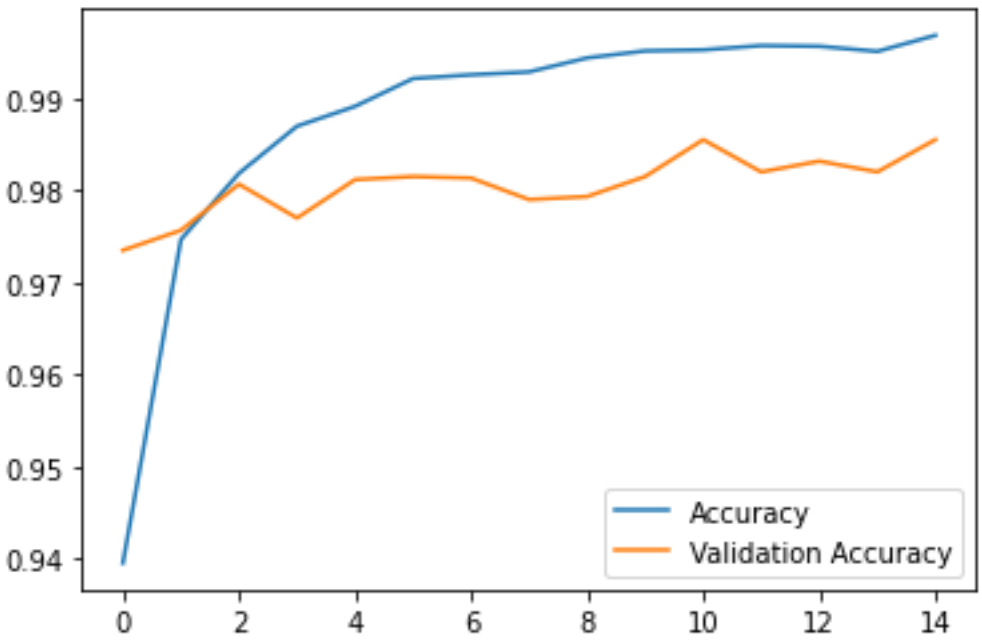
On a bien obtenu les params

615440= 784 neurones *(784 entrees+1BIAS)

78500 =784 neurones *100 tailles de couche +100 tailles de couche

1010 = 100 tailles de couche *10 sorties +10 sorties

3.3



3.3

```
33  #3.1.3
34  m.compile(optimizer="adam", loss="categorical_crossentropy",metrics=["accuracy"])
35
36  #3.2
37  hist = m.fit(train_xdata_flat, train_ydata_onehot, epochs=15 ,validation_split=0.1)
38  plt.plot(hist.epoch, hist.history["accuracy"], label="Accuracy")
39  plt.plot(hist.epoch, hist.history["val_accuracy"], label="Validation Accuracy")
40  plt.legend()
41
42  #3.3
43  accu = m.evaluate(test_xdata_flat,test_ydata_onehot)
44  print("Taux d'erreur est",1-accu[1])
45  print("Taux d'accuracy est",accu[1])
```

```
PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL                    1: Code
1688/1688 [==============================] - 4s 2ms/step - loss: 0.0137 - accuracy: 0.9957 - val_loss: 0.0
Epoch 12/15
1688/1688 [==============================] - 4s 2ms/step - loss: 0.0103 - accuracy: 0.9966 - val_loss: 0.1
Epoch 13/15
1688/1688 [==============================] - 4s 2ms/step - loss: 0.0132 - accuracy: 0.9958 - val_loss: 0.1
Epoch 14/15
1688/1688 [==============================] - 4s 2ms/step - loss: 0.0083 - accuracy: 0.9970 - val_loss: 0.1
Epoch 15/15
1688/1688 [==============================] - 4s 2ms/step - loss: 0.0081 - accuracy: 0.9975 - val_loss: 0.1
313/313 [==============================] - 0s 2ms/step - loss: 0.1201 - accuracy: 0.9782
Taux d'erreur est 0.021799981594085693
Taux d'accuracy est 0.9782000184059143
```

On trouve l'erreur est 2,18%

Apres l'ajout d'une couche de 100 neurones et l'activation relu

Le taux d'erreur baisse à 1,79%