DEEP LEARNING I PROJECT

MASTER OF SCIENCE BIHAR

Subject

PREDICTION OF CLOTHING ITEMS USING DEEP LEARNING METHODES (FASHION MNIST DATASET)

Presented on February 21, 2022 by

BI TOUVOLY ALAIN SERGE DOUDOU
Oumaima ALAMI
Hamza BENTAMMAR

Pr Olivier DUBRULE

SAMMURY

1 Logistic Regression

2 Feed-Forward Neural Network

3 Convolutional Neural Network

4 Transfert learning using RESNET16

5 Conclusion

Logistic Regression

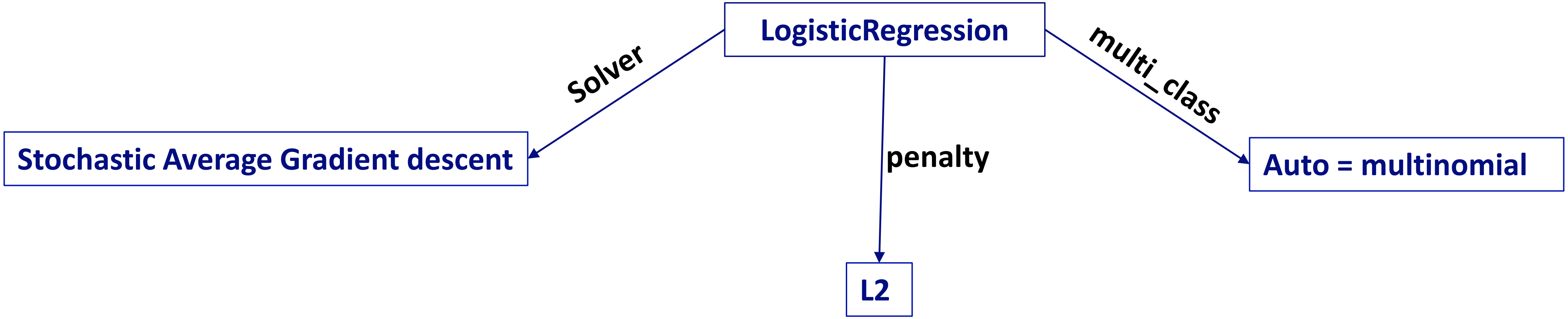
1

Logistic Regression

Model construction

Model Results

Interpretation



LogisticRegression

Solver

Stochastic Average Gradient descent

penalty

L2

multi_class

Auto = multinomial

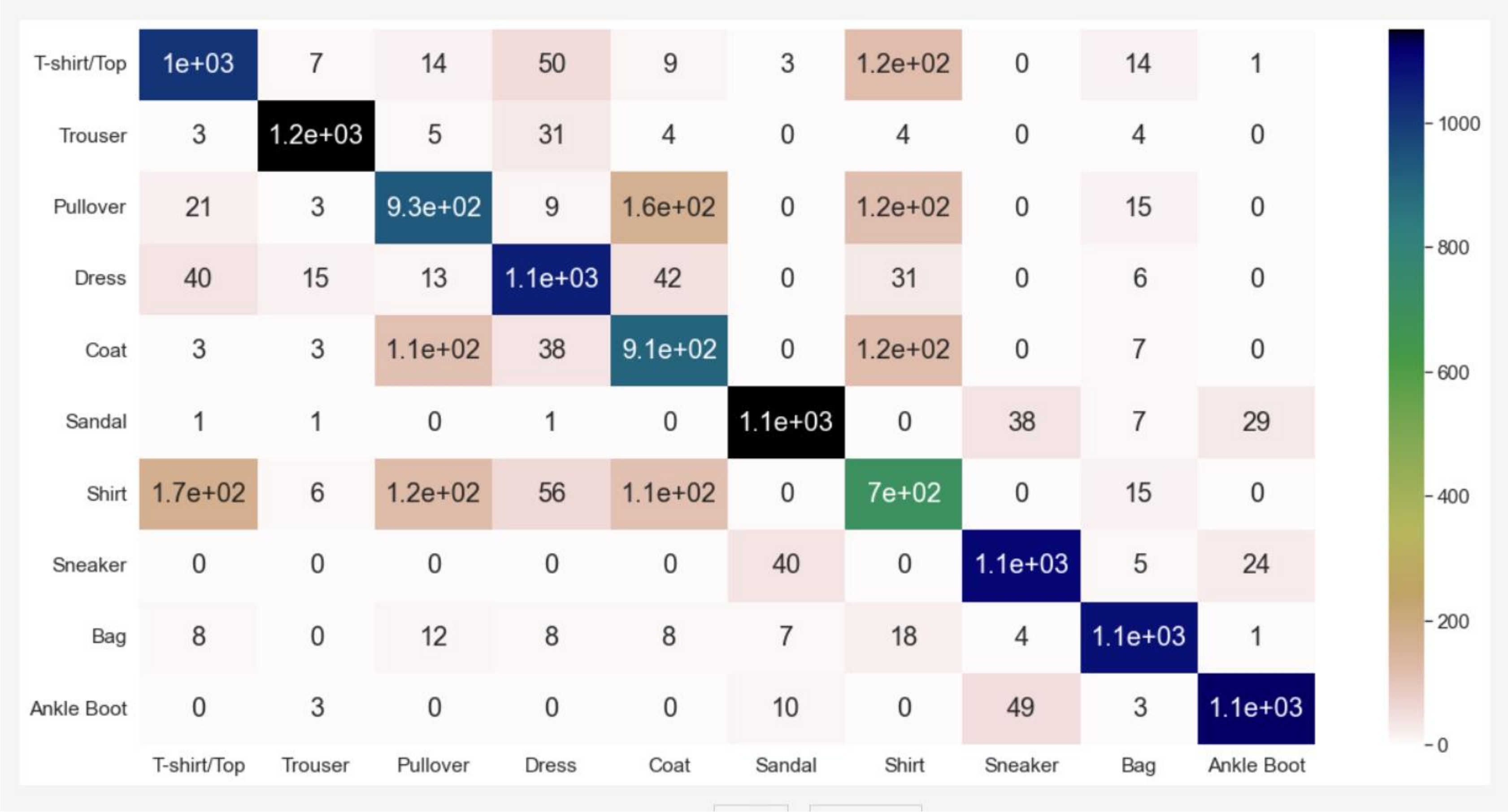
1

Logistic Regression

Model construction

Model Results

Interpretation



	T-shirt/Top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle Boot
T-shirt/Top	1e+03	7	14	50	9	3	1.2e+02	0	14	1
Trouser	3	1.2e+03	5	31	4	0	4	0	4	0
Pullover	21	3	9.3e+02	9	1.6e+02	0	1.2e+02	0	15	0
Dress	40	15	13	1.1e+03	42	0	31	0	6	0
Coat	3	3	1.1e+02	38	9.1e+02	0	1.2e+02	0	7	0
Sandal	1	1	0	1	0	1.1e+03	0	38	7	29
Shirt	1.7e+02	6	1.2e+02	56	1.1e+02	0	7e+02	0	15	0
Sneaker	0	0	0	0	0	40	0	1.1e+03	5	24
Bag	8	0	12	8	8	7	18	4	1.1e+03	1
Ankle Boot	0	3	0	0	0	10	0	49	3	1.1e+03

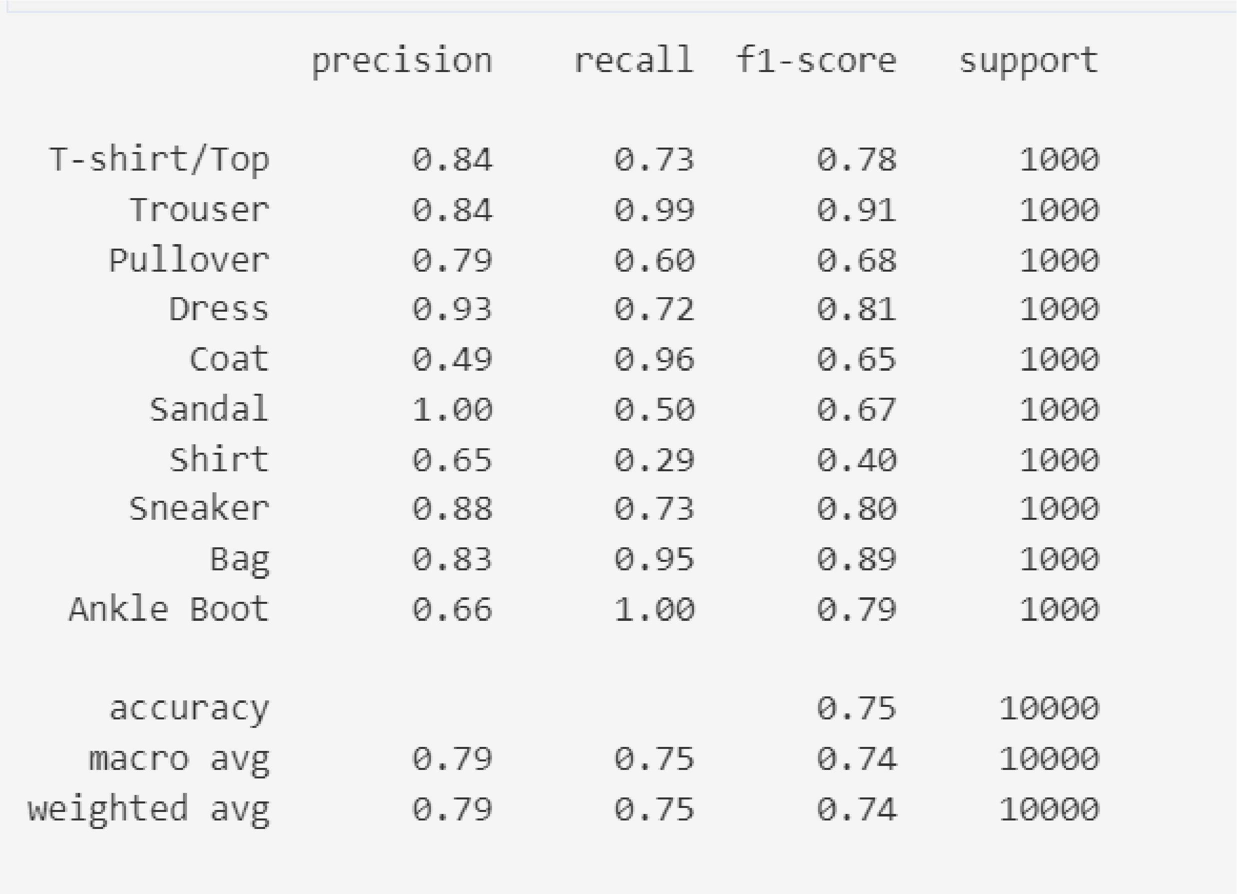
1

Logistic Regression

Model construction

Model Results

Interpretation



```
              precision    recall  f1-score   support

 T-shirt/Top       0.84      0.73      0.78      1000
     Trouser       0.84      0.99      0.91      1000
    Pullover       0.79      0.60      0.68      1000
       Dress       0.93      0.72      0.81      1000
        Coat       0.49      0.96      0.65      1000
      Sandal       1.00      0.50      0.67      1000
       Shirt       0.65      0.29      0.40      1000
     Sneaker       0.88      0.73      0.80      1000
         Bag       0.83      0.95      0.89      1000
  Ankle Boot       0.66      1.00      0.79      1000

    accuracy                           0.75     10000
   macro avg       0.79      0.75      0.74     10000
weighted avg       0.79      0.75      0.74     10000
```

1

Logistic Regression

Model construction

Model Results

Interpretation

Logistic Regression achieved an overall accuracy of **75%** on the Fashion-MNIST dataset.

Looking at the classification report, we can see that the model performs relatively well on some classes, such as class 2 achieving an F1-score of 0.91

However, the model struggles with other classes like class 4 (coat) with F1-score 0.65

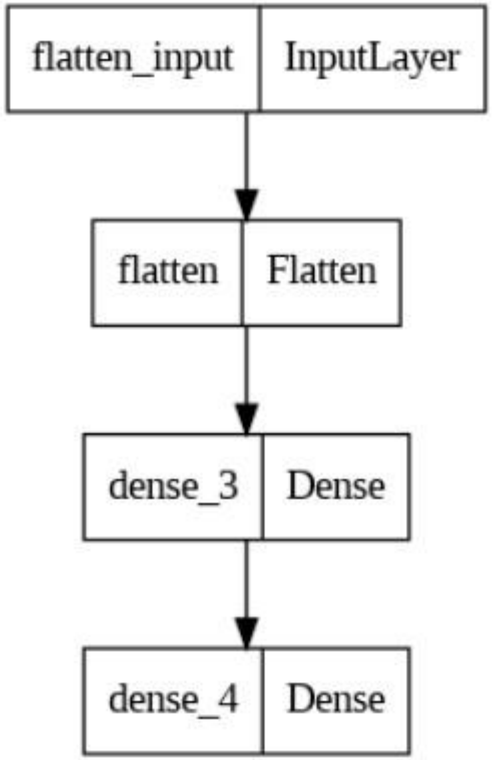
For further improvement we will try the Feed Forward Neural network approach

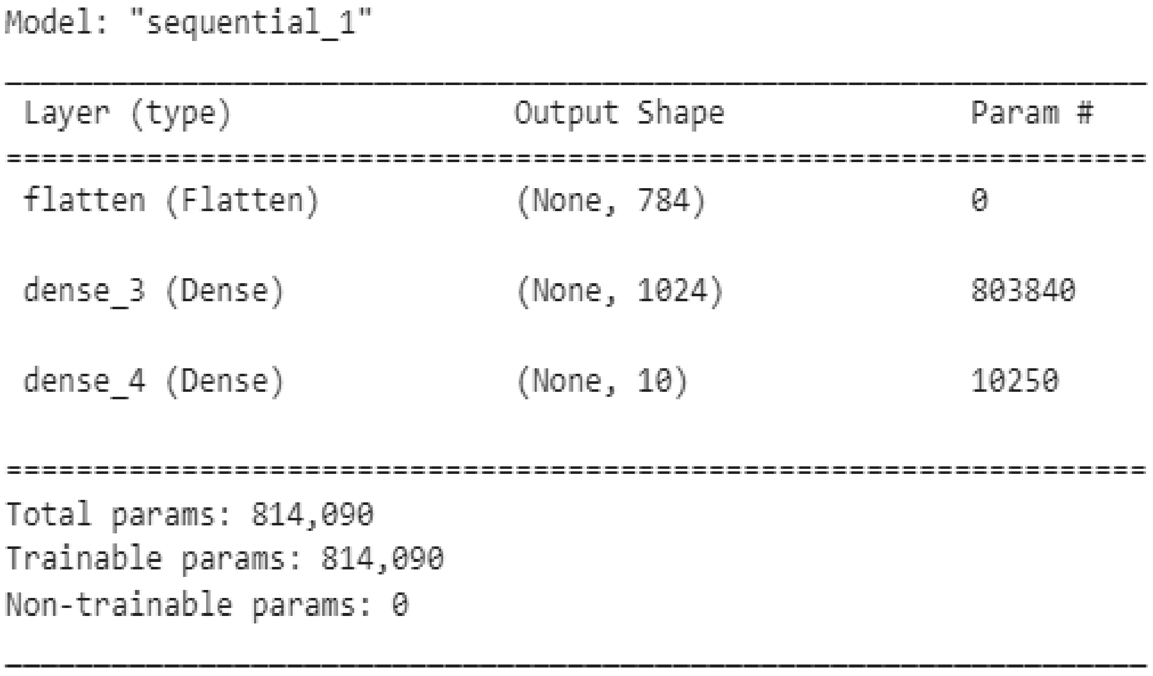
Feed-Forward Neural Network

2

Feed-Forward Neural Network

Model construction — Model Results — Interpretation





```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 784)               0

 dense_3 (Dense)             (None, 1024)              803840

 dense_4 (Dense)             (None, 10)                10250

=================================================================
Total params: 814,090
Trainable params: 814,090
Non-trainable params: 0
_________________________________________________________________
```

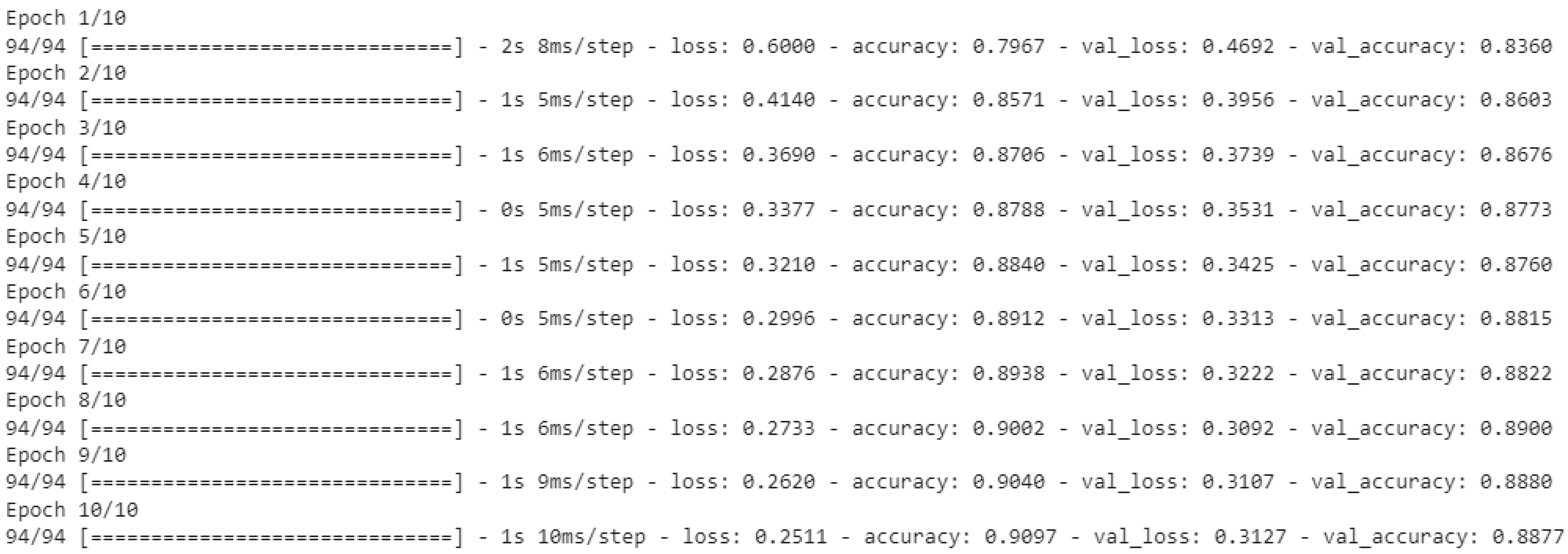
2

Feed-Forward Neural Network

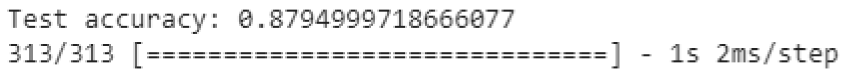
Model construction

Model Results

Interpretation



```
Epoch 1/10
94/94 [==============================] - 2s 8ms/step - loss: 0.6000 - accuracy: 0.7967 - val_loss: 0.4692 - val_accuracy: 0.8360
Epoch 2/10
94/94 [==============================] - 1s 5ms/step - loss: 0.4140 - accuracy: 0.8571 - val_loss: 0.3956 - val_accuracy: 0.8603
Epoch 3/10
94/94 [==============================] - 1s 6ms/step - loss: 0.3690 - accuracy: 0.8706 - val_loss: 0.3739 - val_accuracy: 0.8676
Epoch 4/10
94/94 [==============================] - 0s 5ms/step - loss: 0.3377 - accuracy: 0.8788 - val_loss: 0.3531 - val_accuracy: 0.8773
Epoch 5/10
94/94 [==============================] - 1s 5ms/step - loss: 0.3210 - accuracy: 0.8840 - val_loss: 0.3425 - val_accuracy: 0.8760
Epoch 6/10
94/94 [==============================] - 0s 5ms/step - loss: 0.2996 - accuracy: 0.8912 - val_loss: 0.3313 - val_accuracy: 0.8815
Epoch 7/10
94/94 [==============================] - 1s 6ms/step - loss: 0.2876 - accuracy: 0.8938 - val_loss: 0.3222 - val_accuracy: 0.8822
Epoch 8/10
94/94 [==============================] - 1s 6ms/step - loss: 0.2733 - accuracy: 0.9002 - val_loss: 0.3092 - val_accuracy: 0.8900
Epoch 9/10
94/94 [==============================] - 1s 9ms/step - loss: 0.2620 - accuracy: 0.9040 - val_loss: 0.3107 - val_accuracy: 0.8880
Epoch 10/10
94/94 [==============================] - 1s 10ms/step - loss: 0.2511 - accuracy: 0.9097 - val_loss: 0.3127 - val_accuracy: 0.8877
```



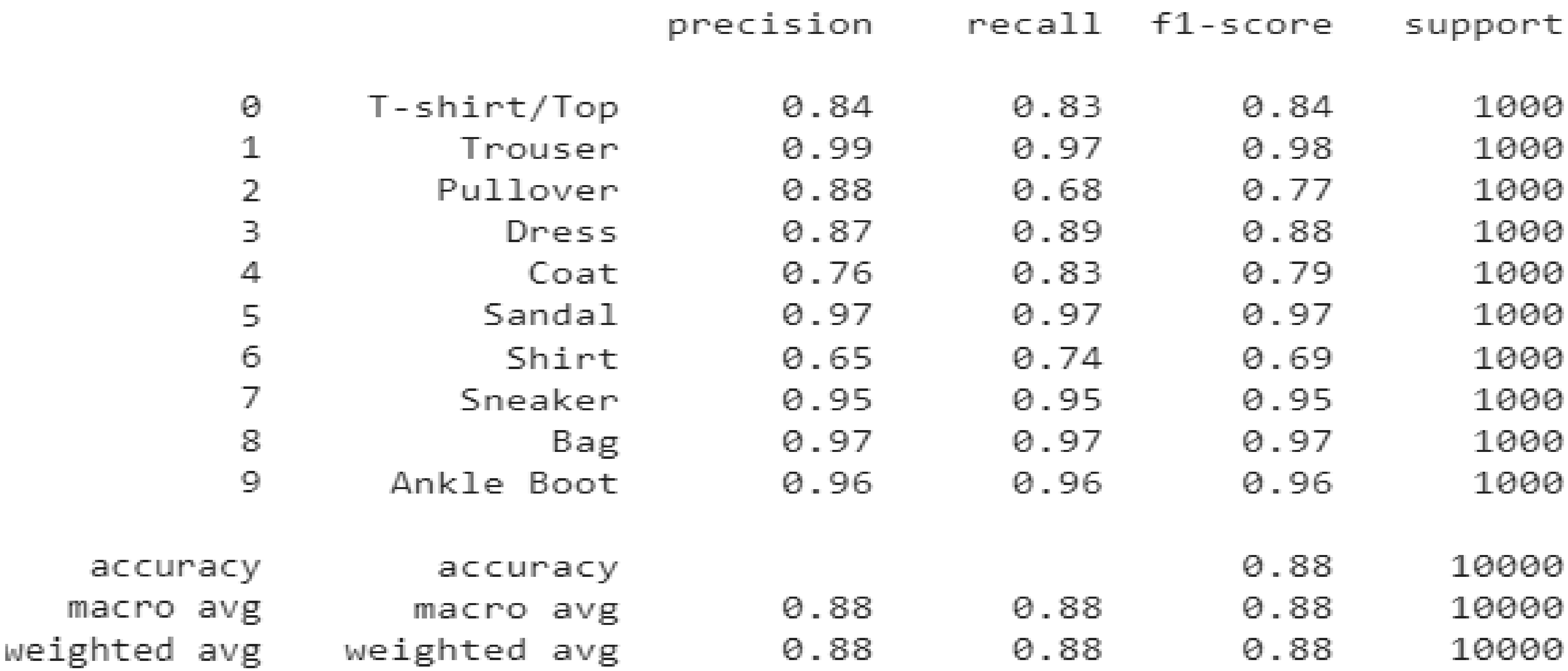
```
Test accuracy: 0.8794999718666077
313/313 [==============================] - 1s 2ms/step
```

2

Feed-Forward Neural Network

Model construction | **Model Results** | Interpretation

Classification Report



```
                               precision    recall  f1-score   support

0          T-shirt/Top           0.84      0.83      0.84      1000
1              Trouser           0.99      0.97      0.98      1000
2             Pullover           0.88      0.68      0.77      1000
3                Dress           0.87      0.89      0.88      1000
4                 Coat           0.76      0.83      0.79      1000
5               Sandal           0.97      0.97      0.97      1000
6                Shirt           0.65      0.74      0.69      1000
7              Sneaker           0.95      0.95      0.95      1000
8                  Bag           0.97      0.97      0.97      1000
9           Ankle Boot           0.96      0.96      0.96      1000

     accuracy             accuracy                         0.88     10000
    macro avg            macro avg       0.88      0.88      0.88     10000
 weighted avg         weighted avg       0.88      0.88      0.88     10000
```

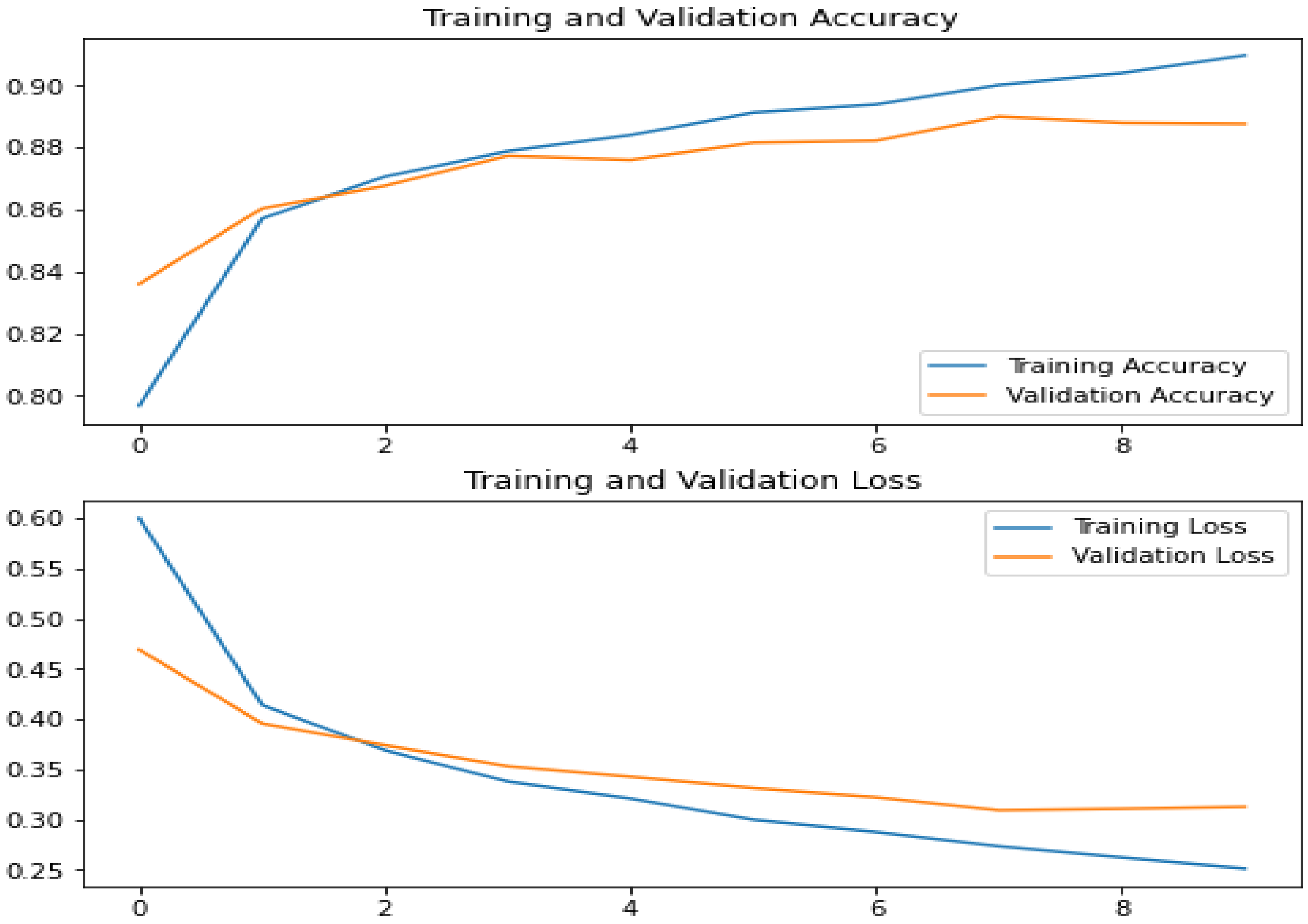
2

Feed-Forward Neural Network

Model construction

Model Results

Interpretation



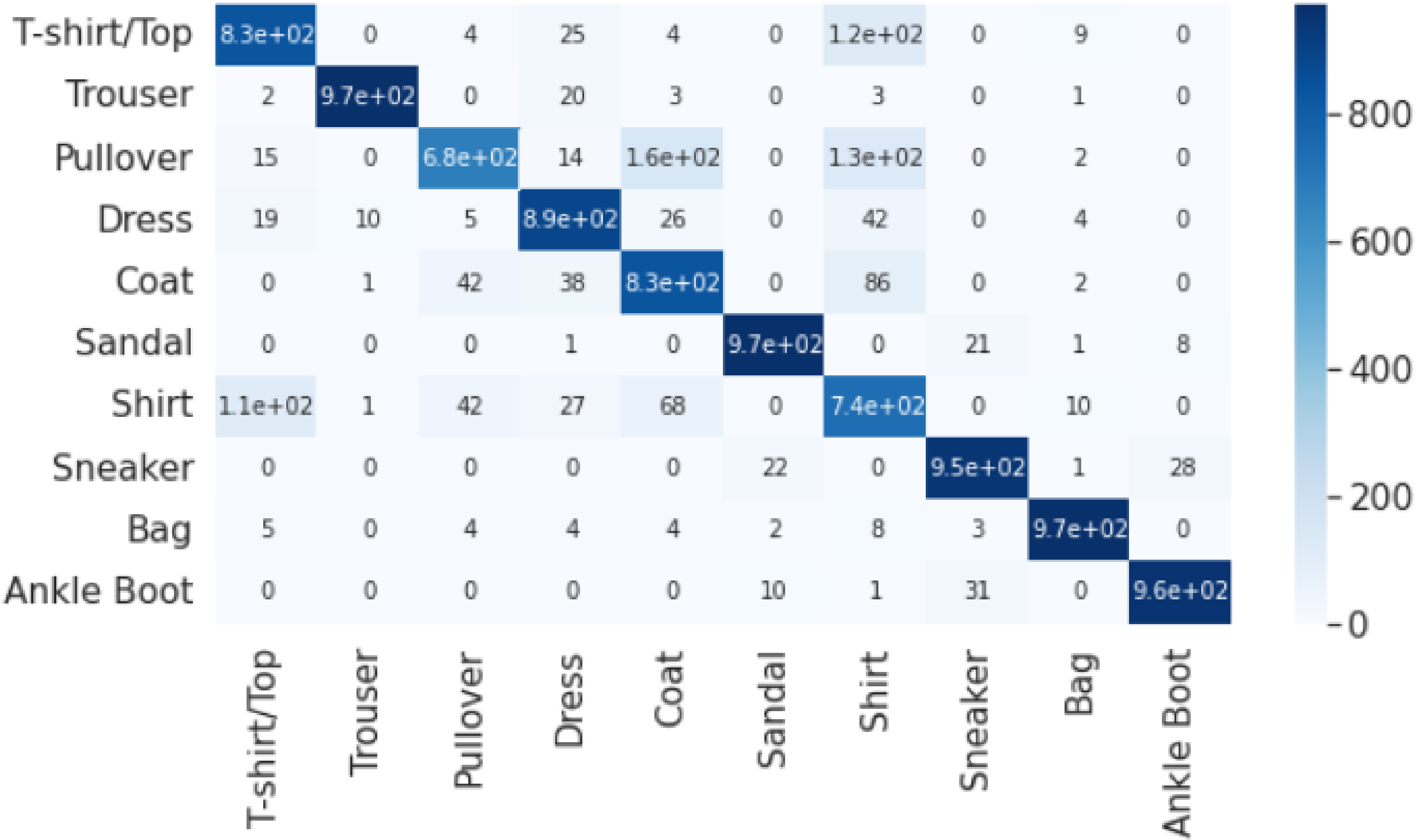
2

Feed-Forward Neural Network

Model construction

Model Results

Interpretation



	T-shirt/Top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle Boot
T-shirt/Top	8.3e+02	0	4	25	4	0	1.2e+02	0	9	0
Trouser	2	9.7e+02	0	20	3	0	3	0	1	0
Pullover	15	0	6.8e+02	14	1.6e+02	0	1.3e+02	0	2	0
Dress	19	10	5	8.9e+02	26	0	42	0	4	0
Coat	0	1	42	38	8.3e+02	0	86	0	2	0
Sandal	0	0	0	1	0	9.7e+02	0	21	1	8
Shirt	1.1e+02	1	42	27	68	0	7.4e+02	0	10	0
Sneaker	0	0	0	0	0	22	0	9.5e+02	1	28
Bag	5	0	4	4	4	2	8	3	9.7e+02	0
Ankle Boot	0	0	0	0	0	10	1	31	0	9.6e+02

2

Feed-Forward Neural Network

Model construction

Model Results

Interpretation

The feedforward neural network (FFNN) achieved an overall accuracy of **87%** on the Fashion-MNIST dataset.

Looking at the classification report, we can see that the model performs relatively well on some classes, such as class 1 (T-shirt/top) and class 5 achieving an F1-score of 0.99 and 0.93, respectively.
However, the model struggles with other classes, such as class 6 (Shirt), achieving an F1-score of 0.65.

For the validation and training accuracy In the case of the FFNN model trained on the Fashion MNIST dataset, **the training accuracy steadily increases and reaches around 93%** by the end of the 10th epoch, while **the validation accuracy reaches a maximum of around 89%** at around the 7th epoch, and then starts to decrease slightly → This suggests that the model ***may be overfitting*** the training data, and may not be generalizing well to the validation set or new data.

Convolutional Neural Network

3

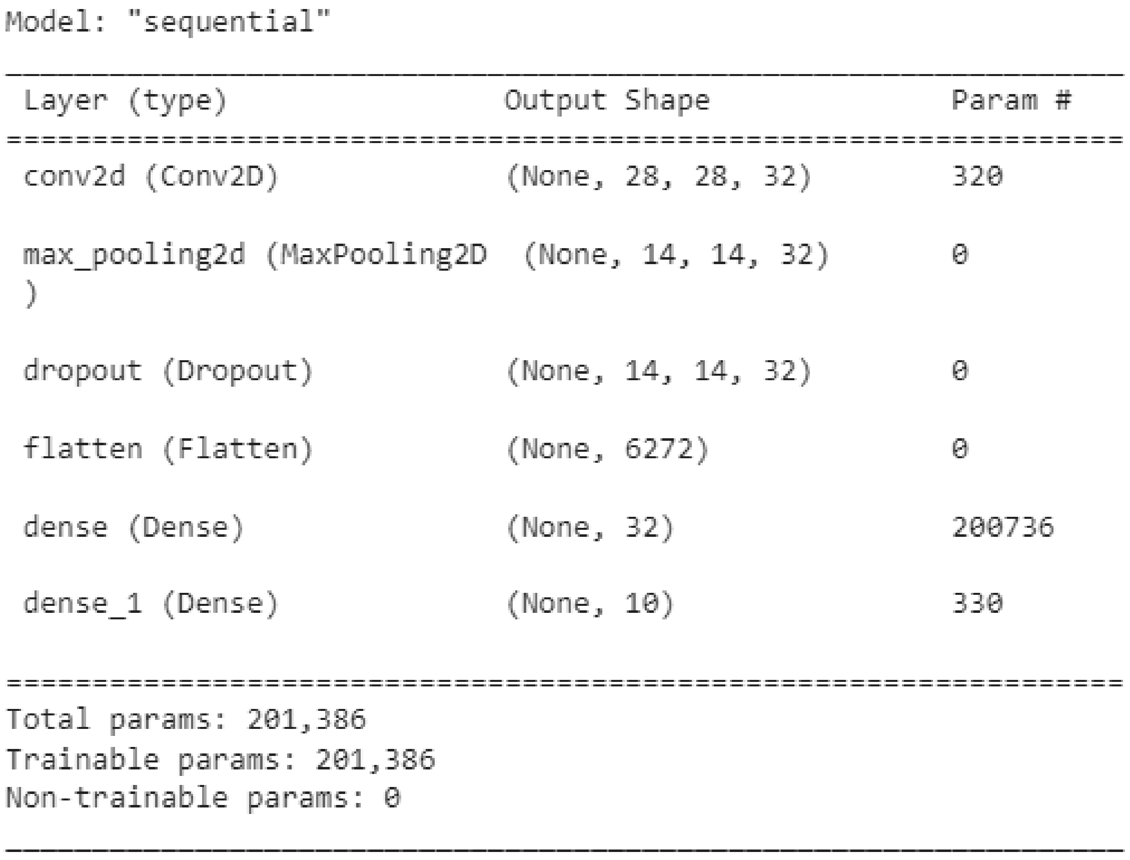
Convolutional Neural Network

First model

Model construction

Model Results

Interpretation



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 32)        320       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 14, 14, 32)       0         
 )                                                               
                                                                 
 dropout (Dropout)           (None, 14, 14, 32)        0         
                                                                 
 flatten (Flatten)           (None, 6272)              0         
                                                                 
 dense (Dense)               (None, 32)                200736    
                                                                 
 dense_1 (Dense)             (None, 10)                330       
                                                                 
=================================================================
Total params: 201,386
Trainable params: 201,386
Non-trainable params: 0
_________________________________________________________________
```

3

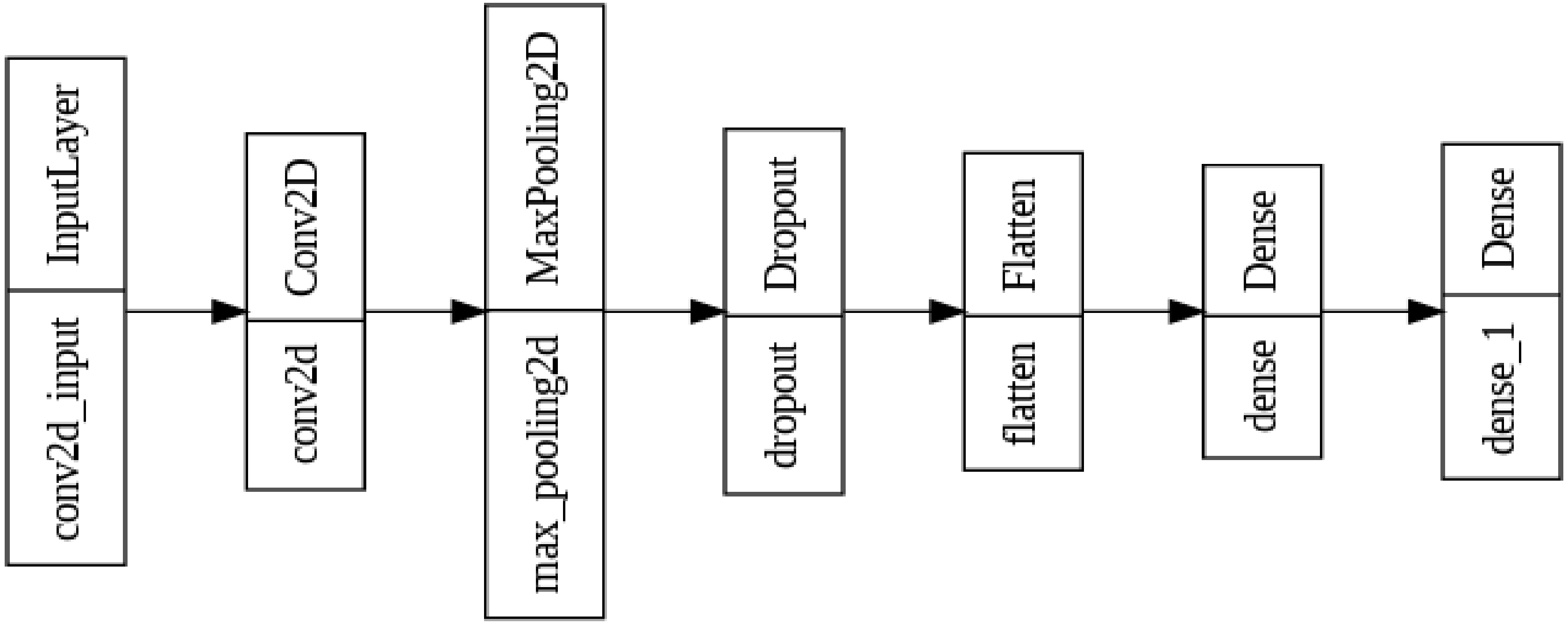
Convolutional Neural Network

First model

Model construction

Model Results

Interpretation



3

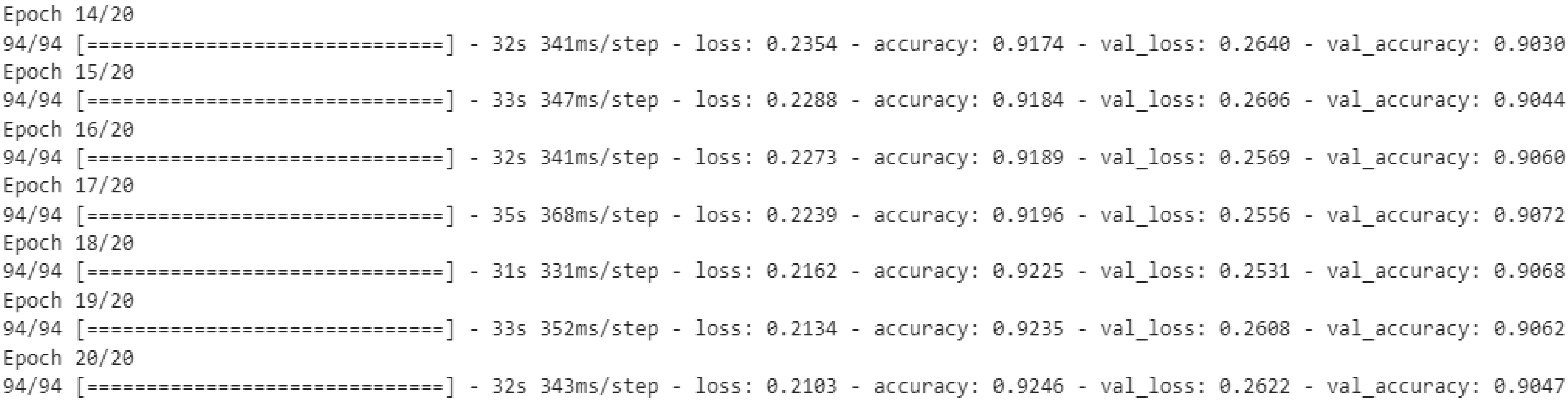
Convolutional Neural Network

First model

Model construction

Model Results

Interpretation



```
Epoch 14/20
94/94 [==============================] - 32s 341ms/step - loss: 0.2354 - accuracy: 0.9174 - val_loss: 0.2640 - val_accuracy: 0.9030
Epoch 15/20
94/94 [==============================] - 33s 347ms/step - loss: 0.2288 - accuracy: 0.9184 - val_loss: 0.2606 - val_accuracy: 0.9044
Epoch 16/20
94/94 [==============================] - 32s 341ms/step - loss: 0.2273 - accuracy: 0.9189 - val_loss: 0.2569 - val_accuracy: 0.9060
Epoch 17/20
94/94 [==============================] - 35s 368ms/step - loss: 0.2239 - accuracy: 0.9196 - val_loss: 0.2556 - val_accuracy: 0.9072
Epoch 18/20
94/94 [==============================] - 31s 331ms/step - loss: 0.2162 - accuracy: 0.9225 - val_loss: 0.2531 - val_accuracy: 0.9068
Epoch 19/20
94/94 [==============================] - 33s 352ms/step - loss: 0.2134 - accuracy: 0.9235 - val_loss: 0.2608 - val_accuracy: 0.9062
Epoch 20/20
94/94 [==============================] - 32s 343ms/step - loss: 0.2103 - accuracy: 0.9246 - val_loss: 0.2622 - val_accuracy: 0.9047
```

3

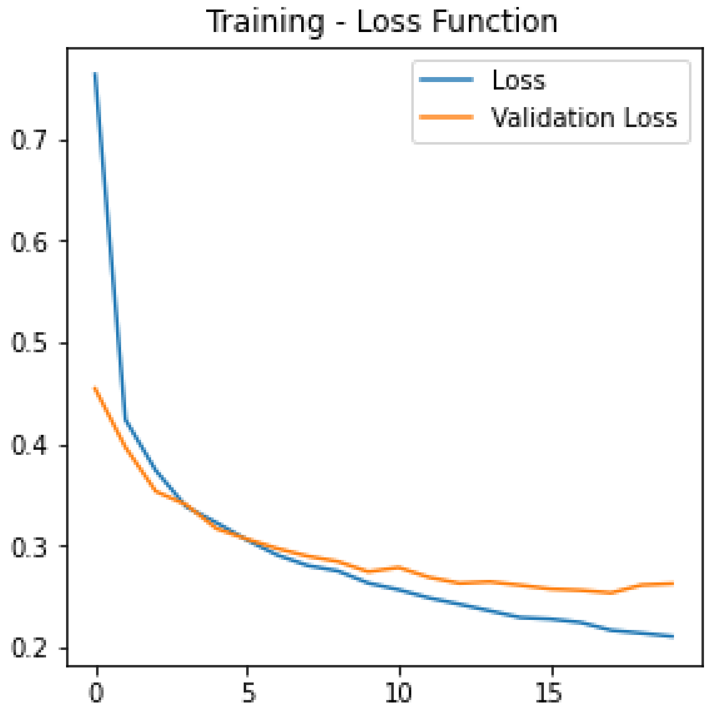
Convolutional Neural Network

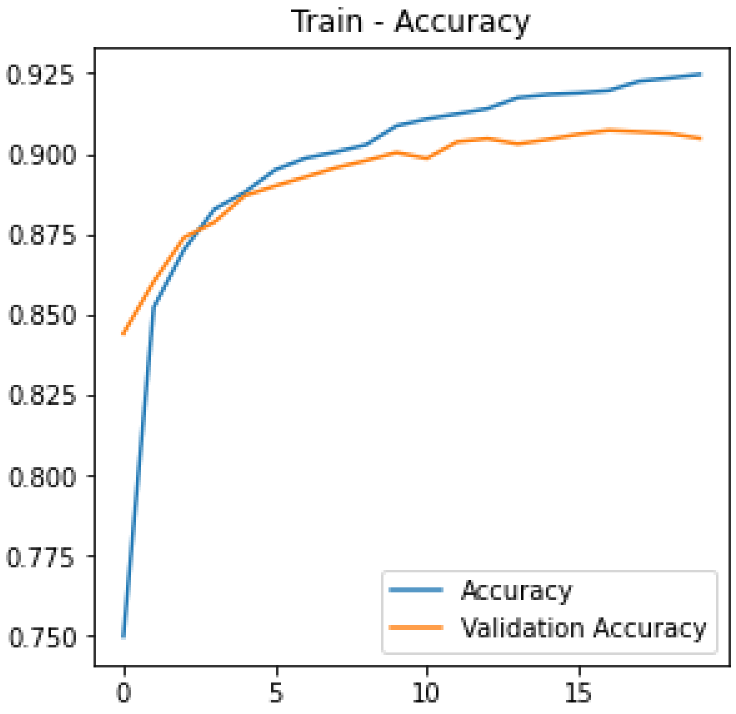
First model

Model construction

Model Results

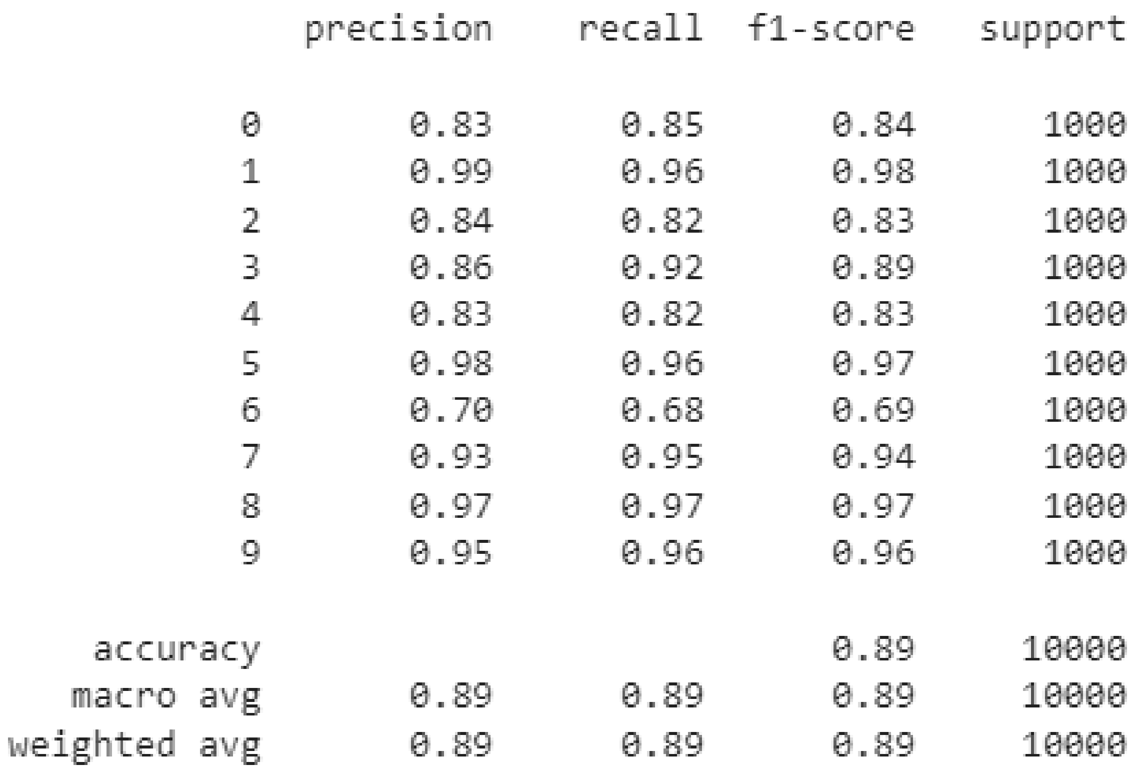
Interpretation


Training - Loss Function


Train - Accuracy

3

Convolutional Neural Network



```
              precision    recall  f1-score   support

           0       0.83      0.85      0.84      1000
           1       0.99      0.96      0.98      1000
           2       0.84      0.82      0.83      1000
           3       0.86      0.92      0.89      1000
           4       0.83      0.82      0.83      1000
           5       0.98      0.96      0.97      1000
           6       0.70      0.68      0.69      1000
           7       0.93      0.95      0.94      1000
           8       0.97      0.97      0.97      1000
           9       0.95      0.96      0.96      1000

    accuracy                           0.89     10000
   macro avg       0.89      0.89      0.89     10000
weighted avg       0.89      0.89      0.89     10000
```

Convolutional Neural Network

The first model of convolutional neural network (CNN) achieved an overall accuracy of **89%** on the Fashion-MNIST dataset.

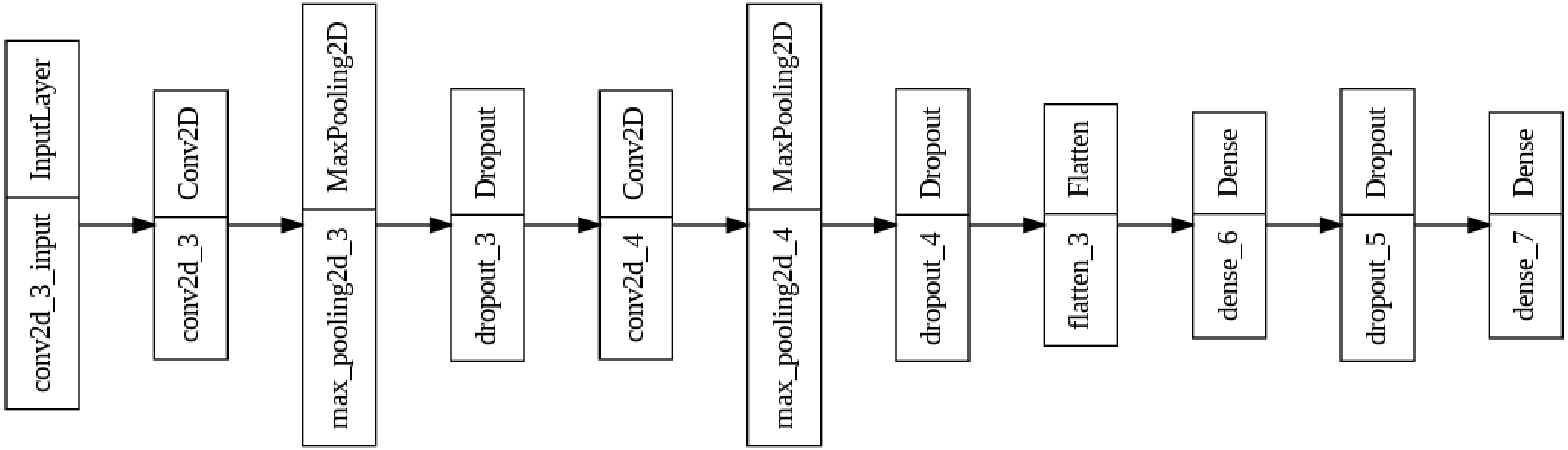
Looking at the classification report, we can see that the model performs relatively well on some classes, such as class 1 and class 5 achieving an F1-score of 0.98 and 0.97, respectively.
However, the model struggles with other classes, such as class 2, achieving an F1-score of 0.83.

Our model begin to overfit after 14 epochs so we use an other model and see results

Convolutional Neural Network

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)               Output Shape              Param #
=================================================================
 conv2d_1 (Conv2D)          (None, 28, 28, 64)        320

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 64)      0
 2D)

 dropout_1 (Dropout)        (None, 14, 14, 64)        0

 conv2d_2 (Conv2D)          (None, 14, 14, 32)        8224

 max_pooling2d_2 (MaxPooling  (None, 7, 7, 32)        0
 2D)

 dropout_2 (Dropout)        (None, 7, 7, 32)          0

 flatten_1 (Flatten)        (None, 1568)              0

 dense_2 (Dense)            (None, 256)               401664

 dropout_3 (Dropout)        (None, 256)               0

 dense_3 (Dense)            (None, 10)                2570

=================================================================
Total params: 412,778
Trainable params: 412,778
Non-trainable params: 0
_________________________________________________________________
```

Convolutional Neural Network



3

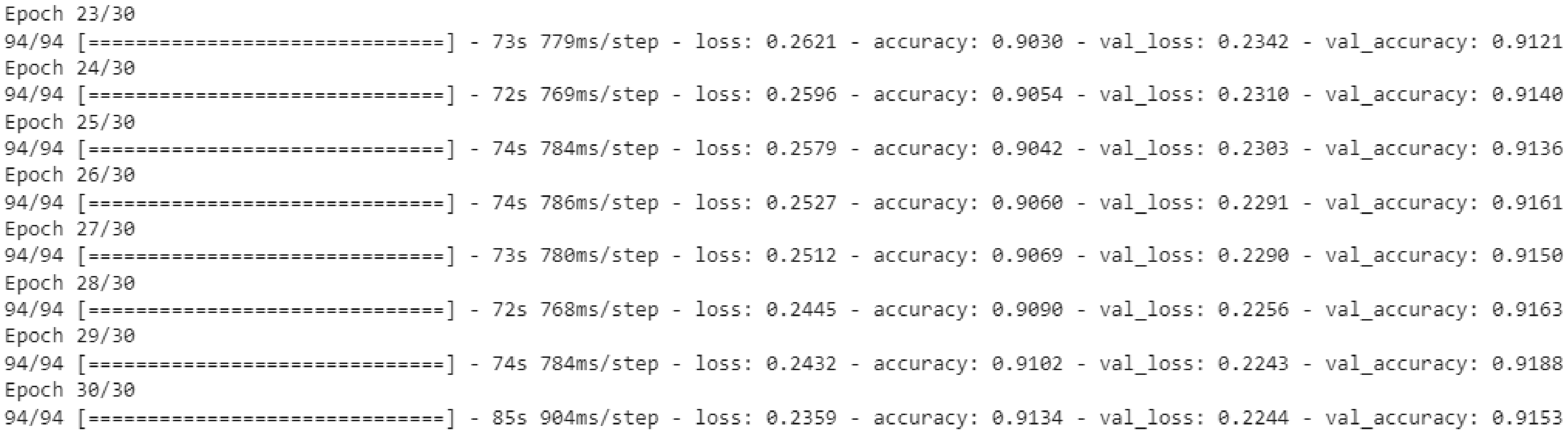
Convolutional Neural Network

second model

Model construction

Model Results

Interpretation



```
Epoch 23/30
94/94 [==============================] - 73s 779ms/step - loss: 0.2621 - accuracy: 0.9030 - val_loss: 0.2342 - val_accuracy: 0.9121
Epoch 24/30
94/94 [==============================] - 72s 769ms/step - loss: 0.2596 - accuracy: 0.9054 - val_loss: 0.2310 - val_accuracy: 0.9140
Epoch 25/30
94/94 [==============================] - 74s 784ms/step - loss: 0.2579 - accuracy: 0.9042 - val_loss: 0.2303 - val_accuracy: 0.9136
Epoch 26/30
94/94 [==============================] - 74s 786ms/step - loss: 0.2527 - accuracy: 0.9060 - val_loss: 0.2291 - val_accuracy: 0.9161
Epoch 27/30
94/94 [==============================] - 73s 780ms/step - loss: 0.2512 - accuracy: 0.9069 - val_loss: 0.2290 - val_accuracy: 0.9150
Epoch 28/30
94/94 [==============================] - 72s 768ms/step - loss: 0.2445 - accuracy: 0.9090 - val_loss: 0.2256 - val_accuracy: 0.9163
Epoch 29/30
94/94 [==============================] - 74s 784ms/step - loss: 0.2432 - accuracy: 0.9102 - val_loss: 0.2243 - val_accuracy: 0.9188
Epoch 30/30
94/94 [==============================] - 85s 904ms/step - loss: 0.2359 - accuracy: 0.9134 - val_loss: 0.2244 - val_accuracy: 0.9153
```

3

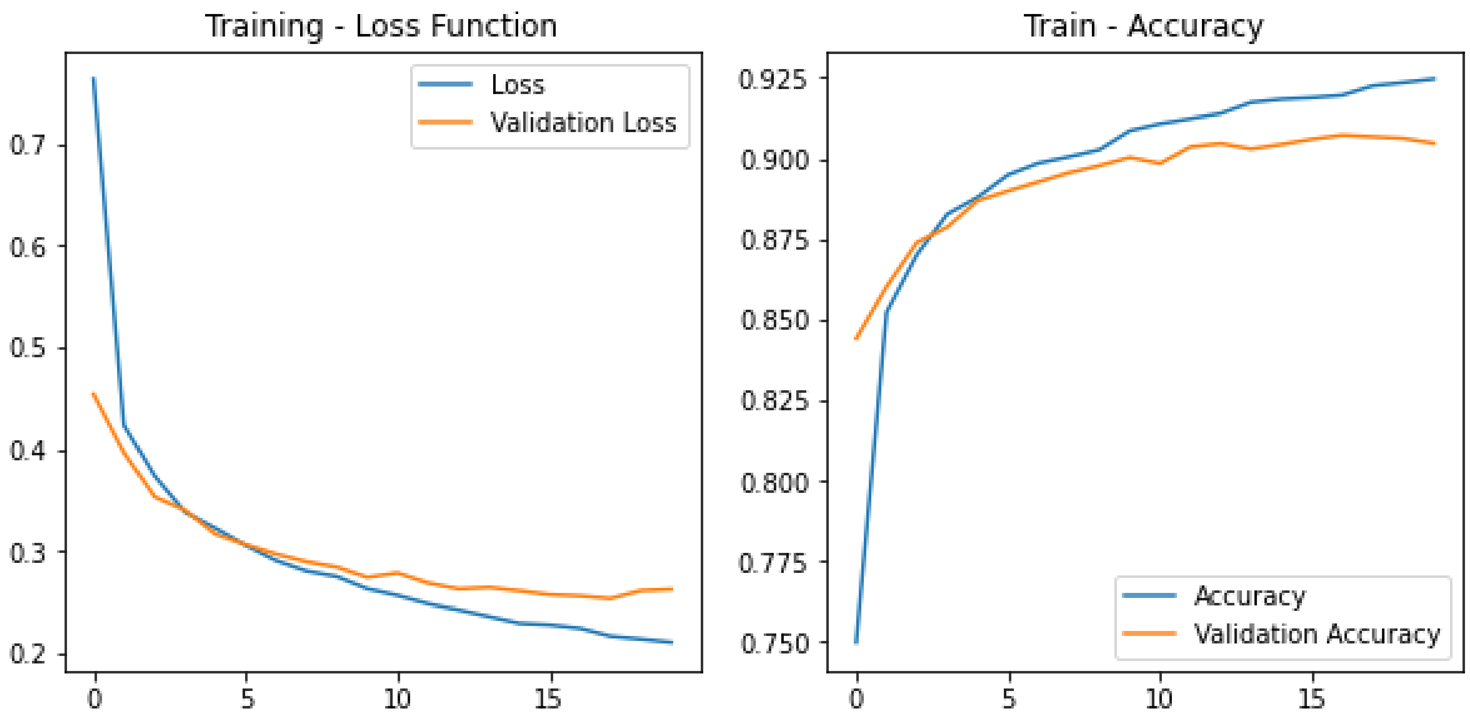
Convolutional Neural Network

second model

Model construction

Model Results

Interpretation



3

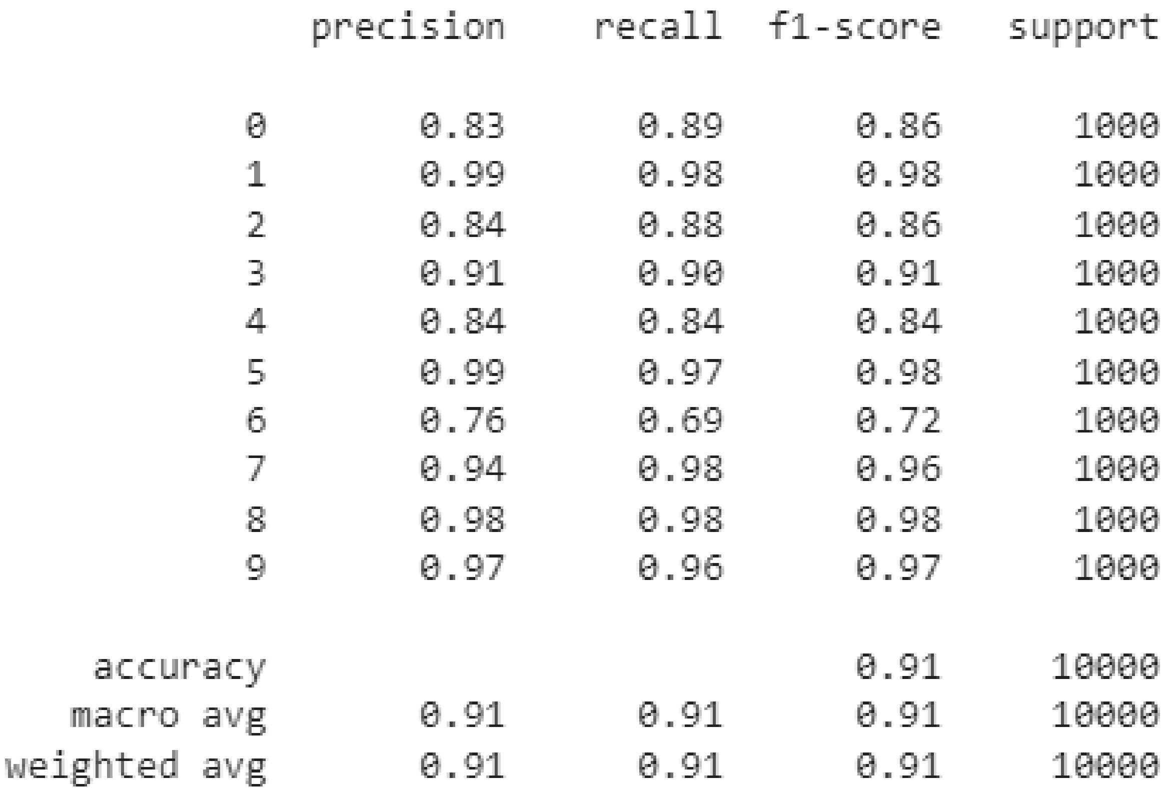
Convolutional Neural Network

second model

Model construction

Model Results

Interpretation



```
              precision    recall  f1-score   support

           0       0.83      0.89      0.86      1000
           1       0.99      0.98      0.98      1000
           2       0.84      0.88      0.86      1000
           3       0.91      0.90      0.91      1000
           4       0.84      0.84      0.84      1000
           5       0.99      0.97      0.98      1000
           6       0.76      0.69      0.72      1000
           7       0.94      0.98      0.96      1000
           8       0.98      0.98      0.98      1000
           9       0.97      0.96      0.97      1000

    accuracy                           0.91     10000
   macro avg       0.91      0.91      0.91     10000
weighted avg       0.91      0.91      0.91     10000
```

Convolutional Neural Network

The second model of convolutional neural network (CNN) achieved an overall accuracy of **91%** on the Fashion-MNIST dataset.

Looking at the classification report, we can see that the model performs very well on some classes, such as class 1 and class 5 and class 8 achieving an F1-score of 0.98.

However, the model struggles with other classes, such as class 6, achieving an F1-score of 0.72.

The second model gives as a better result but we will try to find the best hyper-parameters.

3

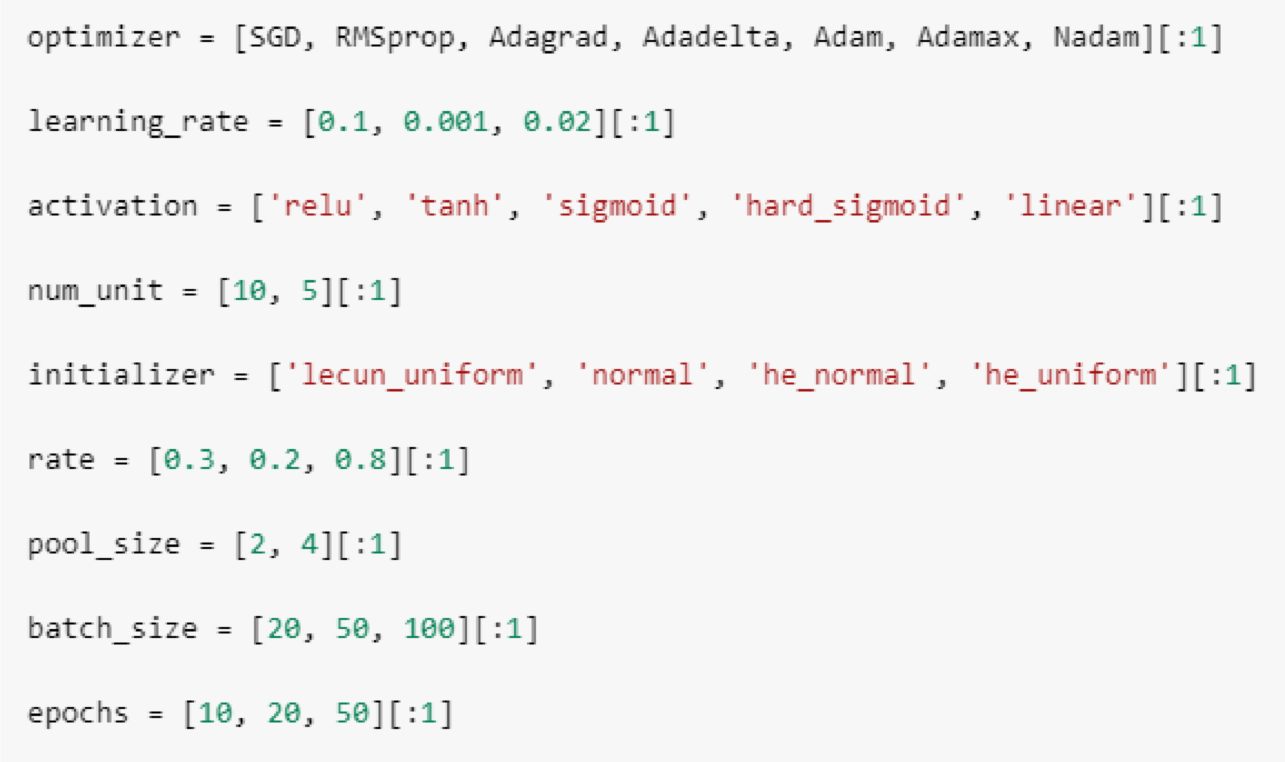
Convolutional Neural Network

Best Hyper-Parameters

Best parameters

Model Results

Interpretation



```
optimizer = [SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, Nadam][:1]

learning_rate = [0.1, 0.001, 0.02][:1]

activation = ['relu', 'tanh', 'sigmoid', 'hard_sigmoid', 'linear'][:1]

num_unit = [10, 5][:1]

initializer = ['lecun_uniform', 'normal', 'he_normal', 'he_uniform'][:1]

rate = [0.3, 0.2, 0.8][:1]

pool_size = [2, 4][:1]

batch_size = [20, 50, 100][:1]

epochs = [10, 20, 50][:1]
```

3

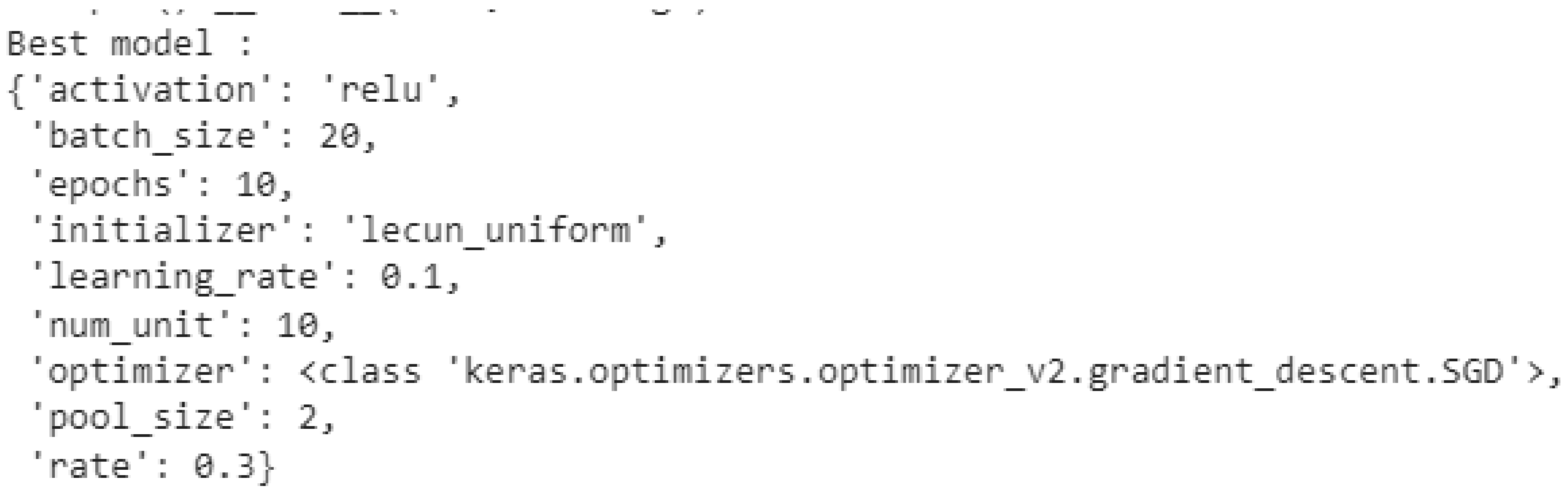
Convolutional Neural Network

Best Hyper-Parameters

Best parameters

Model Results

Interpretation

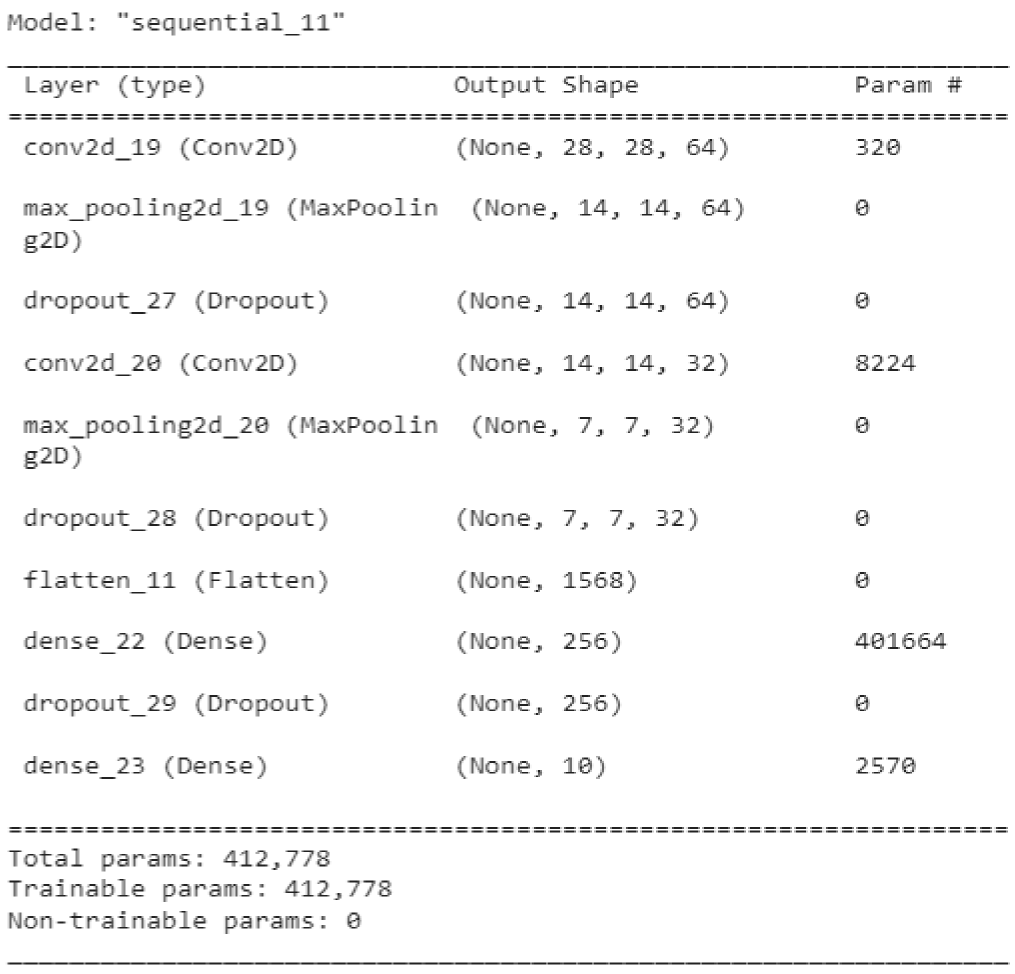


```
Best model :
{'activation': 'relu',
 'batch_size': 20,
 'epochs': 10,
 'initializer': 'lecun_uniform',
 'learning_rate': 0.1,
 'num_unit': 10,
 'optimizer': <class 'keras.optimizers.optimizer_v2.gradient_descent.SGD'>,
 'pool_size': 2,
 'rate': 0.3}
```

3

Convolutional Neural Network

Best Hyper-Parameters | Best parameters | Model Results | Interpretation



```
Model: "sequential_11"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_19 (Conv2D)          (None, 28, 28, 64)        320       
                                                                 
 max_pooling2d_19 (MaxPoolin  (None, 14, 14, 64)       0         
 g2D)                                                            
                                                                 
 dropout_27 (Dropout)        (None, 14, 14, 64)        0         
                                                                 
 conv2d_20 (Conv2D)          (None, 14, 14, 32)        8224      
                                                                 
 max_pooling2d_20 (MaxPoolin  (None, 7, 7, 32)         0         
 g2D)                                                            
                                                                 
 dropout_28 (Dropout)        (None, 7, 7, 32)          0         
                                                                 
 flatten_11 (Flatten)        (None, 1568)              0         
                                                                 
 dense_22 (Dense)            (None, 256)               401664    
                                                                 
 dropout_29 (Dropout)        (None, 256)               0         
                                                                 
 dense_23 (Dense)            (None, 10)                2570      
                                                                 
=================================================================
Total params: 412,778
Trainable params: 412,778
Non-trainable params: 0
_________________________________________________________________
```

3

Convolutional Neural Network

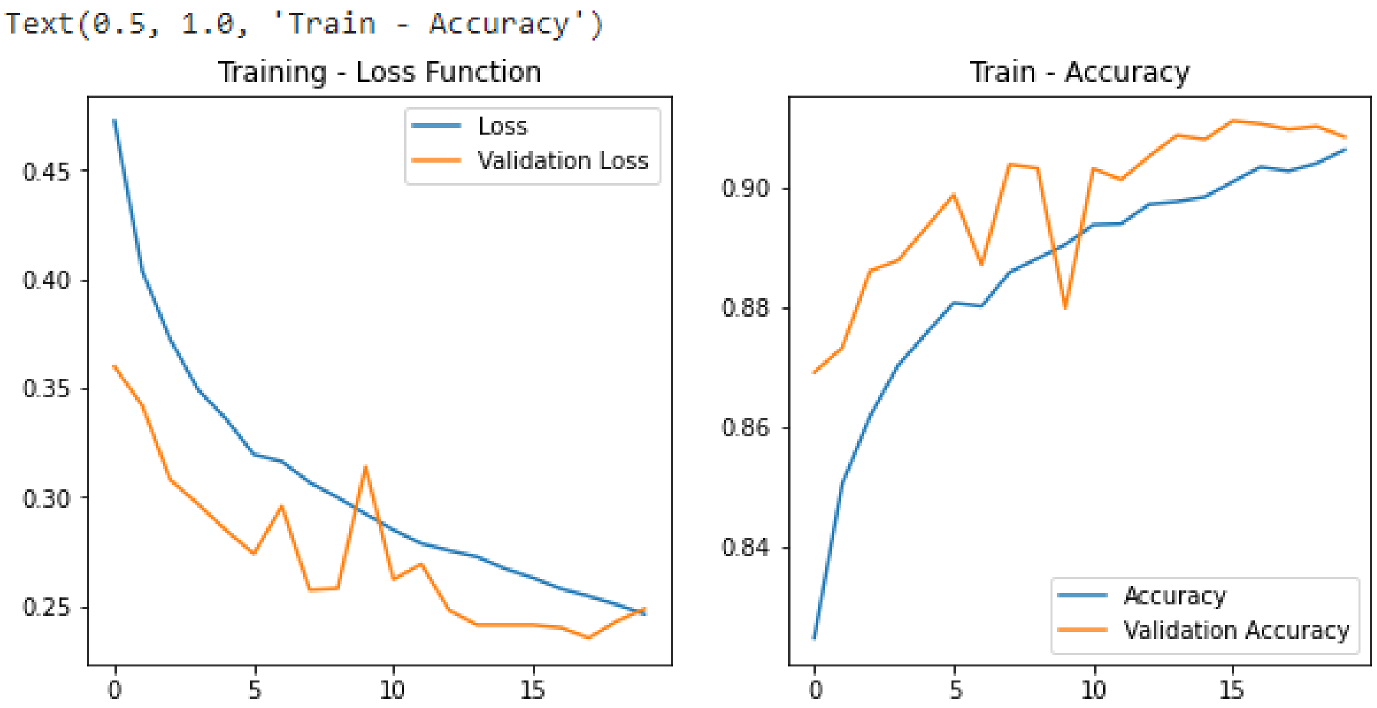
Best Hyper-Parameters

Best parameters

Model Results

Interpretation

▼



```
Text(0.5, 1.0, 'Train - Accuracy')
```

3

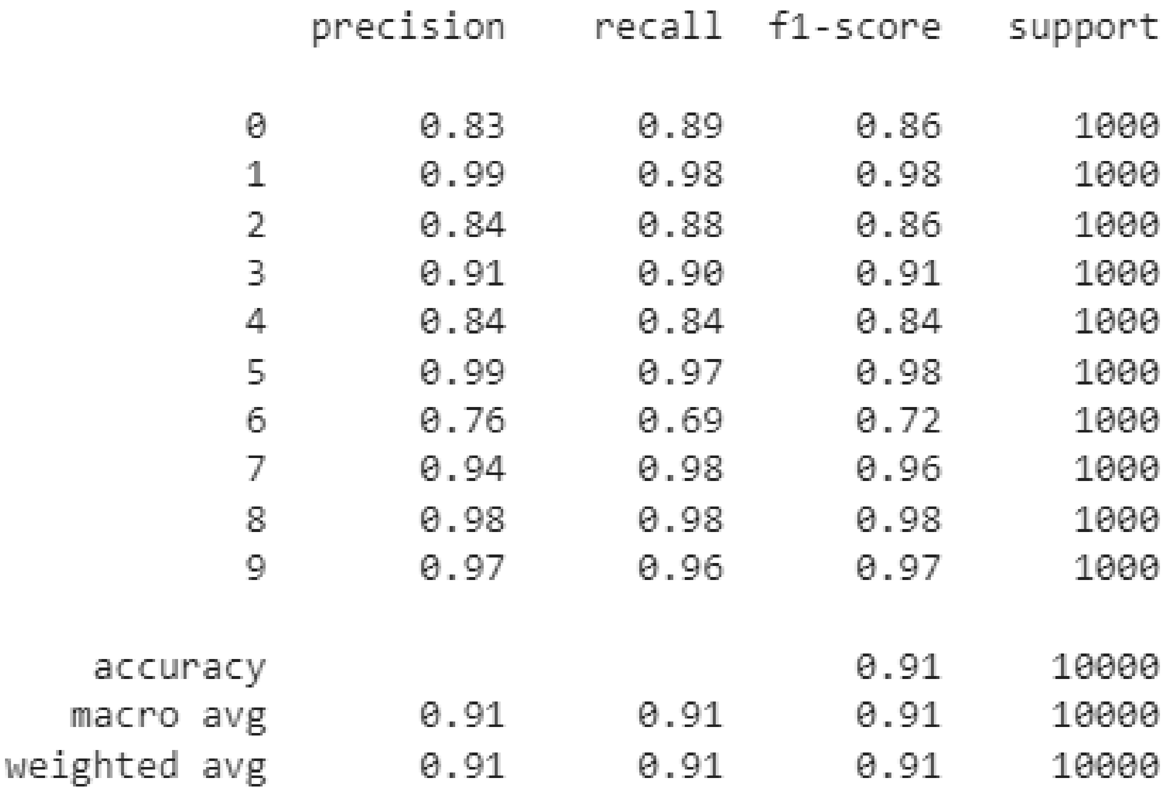
Convolutional Neural Network

Best Hyper-Parameters

Best parameters

Model Results

Interpretation



```
              precision    recall  f1-score   support

           0       0.83      0.89      0.86      1000
           1       0.99      0.98      0.98      1000
           2       0.84      0.88      0.86      1000
           3       0.91      0.90      0.91      1000
           4       0.84      0.84      0.84      1000
           5       0.99      0.97      0.98      1000
           6       0.76      0.69      0.72      1000
           7       0.94      0.98      0.96      1000
           8       0.98      0.98      0.98      1000
           9       0.97      0.96      0.97      1000

    accuracy                           0.91     10000
   macro avg       0.91      0.91      0.91     10000
weighted avg       0.91      0.91      0.91     10000
```

Convolutional Neural Network

▼

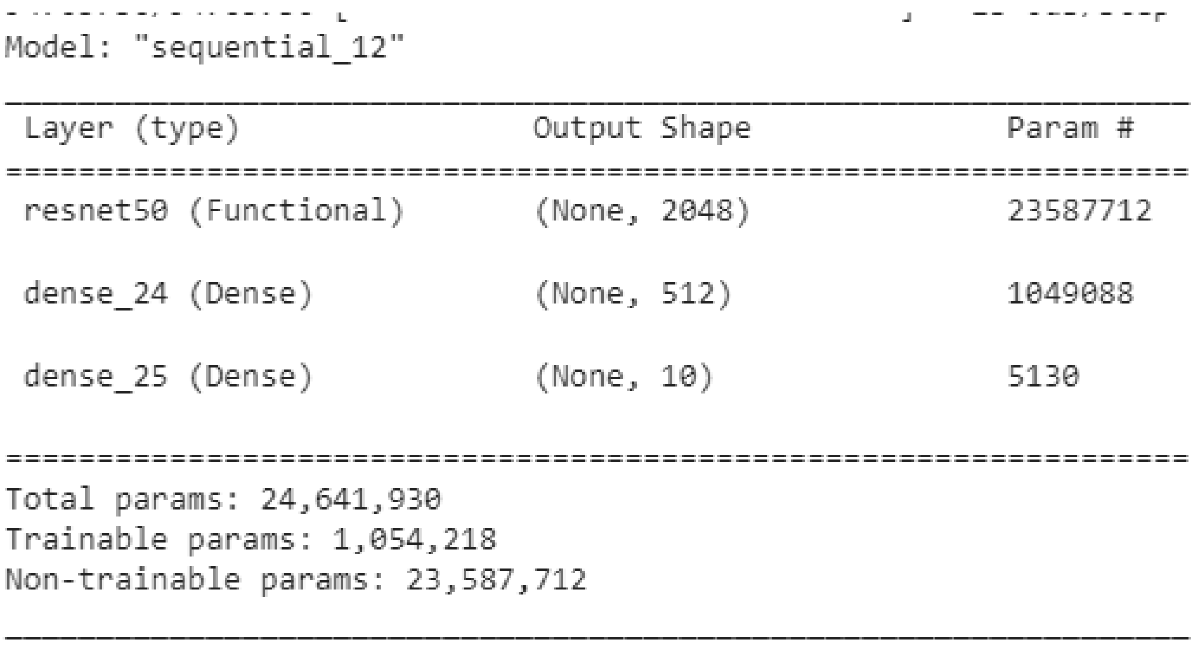
After running the the model with best parameters, values doesn’t change a lot.
We achieve an overall accuracy of **91%** on the Fashion-MNIST dataset.

Transfert learning using RESNET50

4

Transfert learning using RESNET50

Model construction | Model Results | Interpretation



```
Model: "sequential_12"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 2048)              23587712

 dense_24 (Dense)            (None, 512)               1049088

 dense_25 (Dense)            (None, 10)                5130

=================================================================
Total params: 24,641,930
Trainable params: 1,054,218
Non-trainable params: 23,587,712
_________________________________________________________________
```

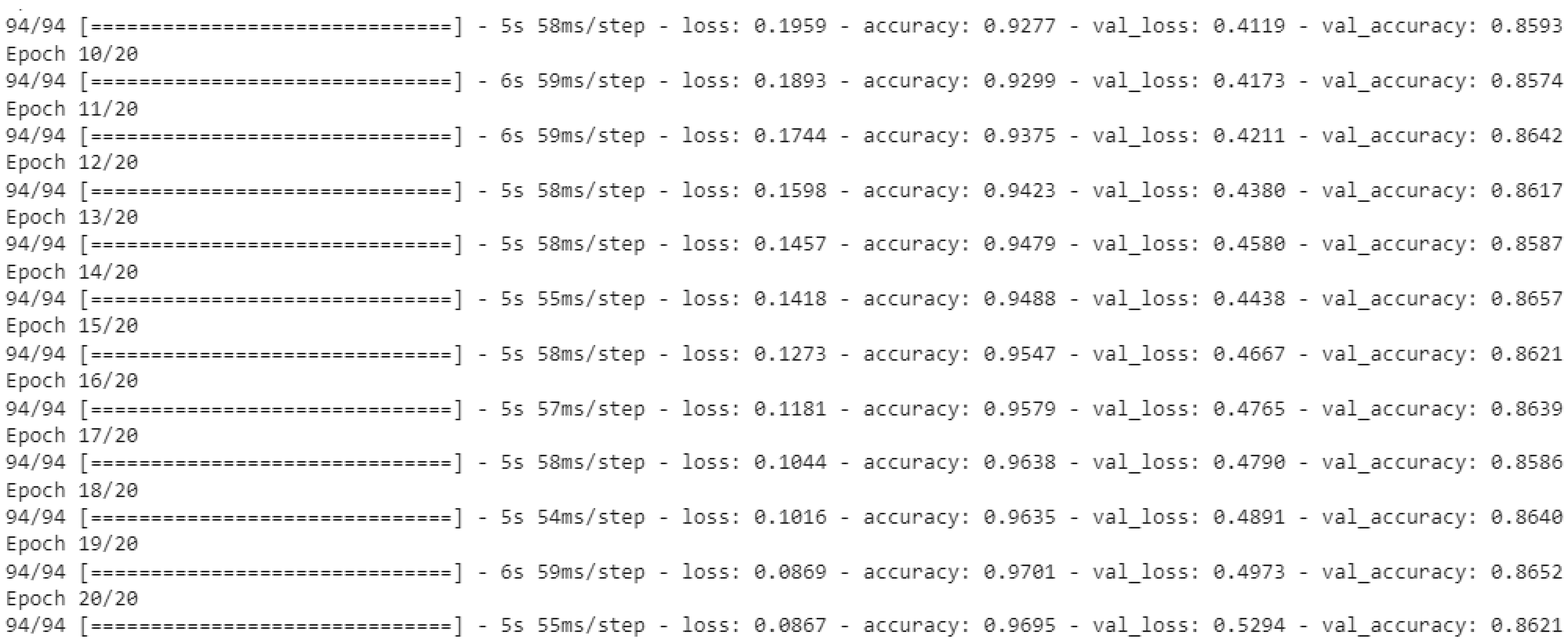
4

Transfert learning using RESNET50

Model construction

Model Results

Interpretation



```
94/94 [==============================] - 5s 58ms/step - loss: 0.1959 - accuracy: 0.9277 - val_loss: 0.4119 - val_accuracy: 0.8593
Epoch 10/20
94/94 [==============================] - 6s 59ms/step - loss: 0.1893 - accuracy: 0.9299 - val_loss: 0.4173 - val_accuracy: 0.8574
Epoch 11/20
94/94 [==============================] - 6s 59ms/step - loss: 0.1744 - accuracy: 0.9375 - val_loss: 0.4211 - val_accuracy: 0.8642
Epoch 12/20
94/94 [==============================] - 5s 58ms/step - loss: 0.1598 - accuracy: 0.9423 - val_loss: 0.4380 - val_accuracy: 0.8617
Epoch 13/20
94/94 [==============================] - 5s 58ms/step - loss: 0.1457 - accuracy: 0.9479 - val_loss: 0.4580 - val_accuracy: 0.8587
Epoch 14/20
94/94 [==============================] - 5s 55ms/step - loss: 0.1418 - accuracy: 0.9488 - val_loss: 0.4438 - val_accuracy: 0.8657
Epoch 15/20
94/94 [==============================] - 5s 58ms/step - loss: 0.1273 - accuracy: 0.9547 - val_loss: 0.4667 - val_accuracy: 0.8621
Epoch 16/20
94/94 [==============================] - 5s 57ms/step - loss: 0.1181 - accuracy: 0.9579 - val_loss: 0.4765 - val_accuracy: 0.8639
Epoch 17/20
94/94 [==============================] - 5s 58ms/step - loss: 0.1044 - accuracy: 0.9638 - val_loss: 0.4790 - val_accuracy: 0.8586
Epoch 18/20
94/94 [==============================] - 5s 54ms/step - loss: 0.1016 - accuracy: 0.9635 - val_loss: 0.4891 - val_accuracy: 0.8640
Epoch 19/20
94/94 [==============================] - 6s 59ms/step - loss: 0.0869 - accuracy: 0.9701 - val_loss: 0.4973 - val_accuracy: 0.8652
Epoch 20/20
94/94 [==============================] - 5s 55ms/step - loss: 0.0867 - accuracy: 0.9695 - val_loss: 0.5294 - val_accuracy: 0.8621
```

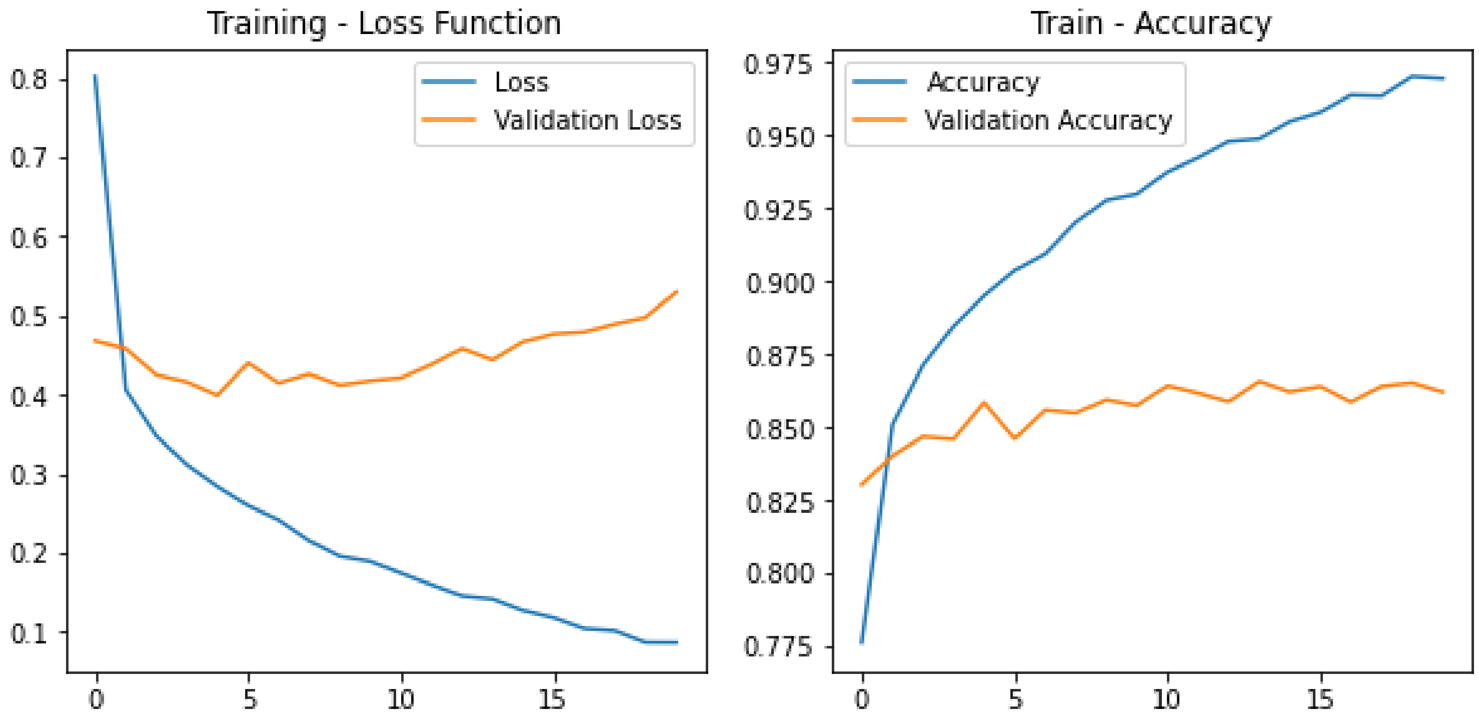
4

Transfert learning using RESNET16

Model construction

Model Results

Interpretation



4

Transfert learning using RESNET50

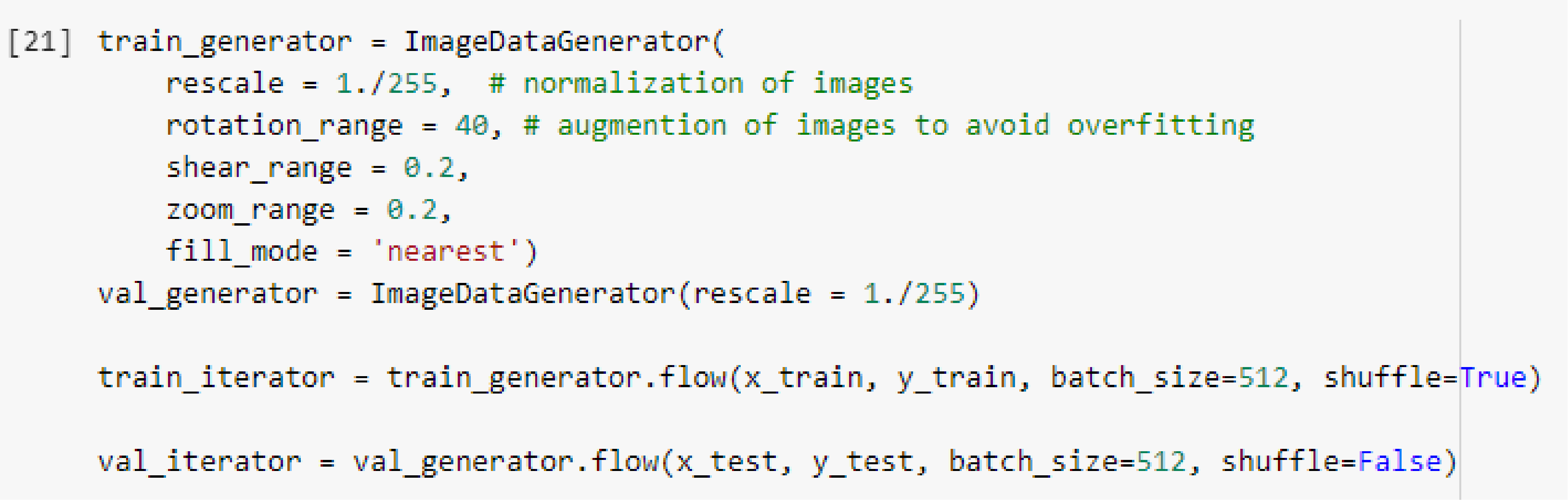
Model construction

Model Results

Interpretation

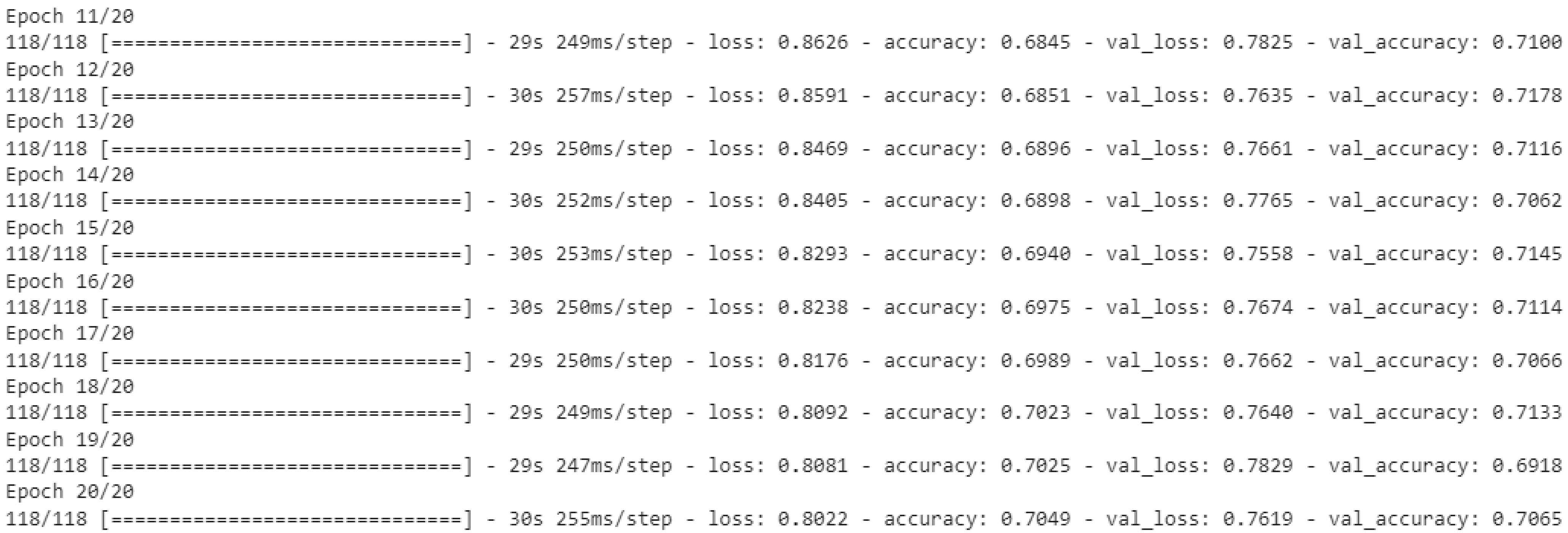
- ResNet is a model trained by RBG images and has over 23 millions parameters so it’s natural that the model will overfit with our data (grey scale images)
- We will try to do data augmentation in order to help us to deal with this overfitting problem

Transfert learning using RESNET50



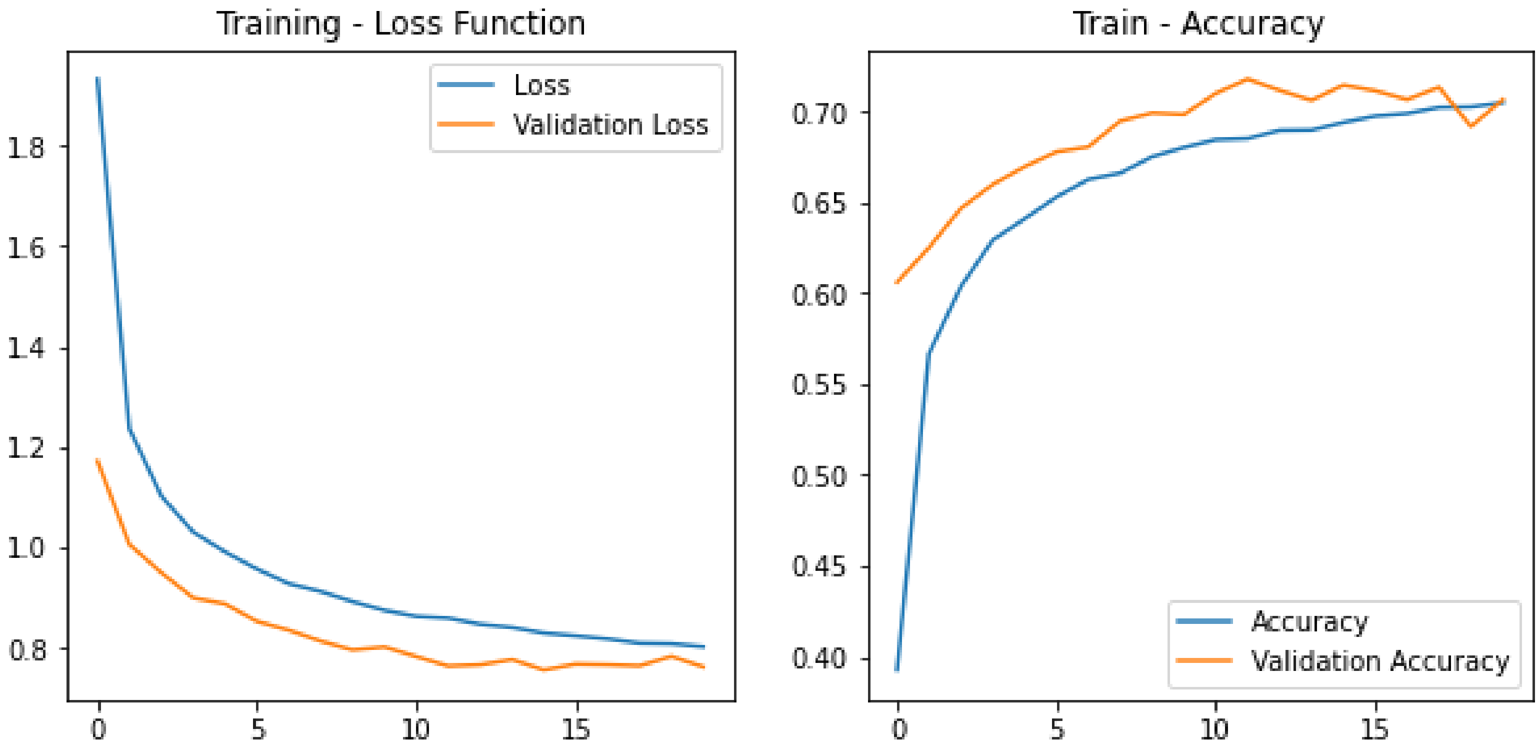
```
[21] train_generator = ImageDataGenerator(
         rescale = 1./255,  # normalization of images
         rotation_range = 40, # augmention of images to avoid overfitting
         shear_range = 0.2,
         zoom_range = 0.2,
         fill_mode = 'nearest')
     val_generator = ImageDataGenerator(rescale = 1./255)

     train_iterator = train_generator.flow(x_train, y_train, batch_size=512, shuffle=True)

     val_iterator = val_generator.flow(x_test, y_test, batch_size=512, shuffle=False)
```

Transfert learning using RESNET50



```
Epoch 11/20
118/118 [==============================] - 29s 249ms/step - loss: 0.8626 - accuracy: 0.6845 - val_loss: 0.7825 - val_accuracy: 0.7100
Epoch 12/20
118/118 [==============================] - 30s 257ms/step - loss: 0.8591 - accuracy: 0.6851 - val_loss: 0.7635 - val_accuracy: 0.7178
Epoch 13/20
118/118 [==============================] - 29s 250ms/step - loss: 0.8469 - accuracy: 0.6896 - val_loss: 0.7661 - val_accuracy: 0.7116
Epoch 14/20
118/118 [==============================] - 30s 252ms/step - loss: 0.8405 - accuracy: 0.6898 - val_loss: 0.7765 - val_accuracy: 0.7062
Epoch 15/20
118/118 [==============================] - 30s 253ms/step - loss: 0.8293 - accuracy: 0.6940 - val_loss: 0.7558 - val_accuracy: 0.7145
Epoch 16/20
118/118 [==============================] - 30s 250ms/step - loss: 0.8238 - accuracy: 0.6975 - val_loss: 0.7674 - val_accuracy: 0.7114
Epoch 17/20
118/118 [==============================] - 29s 250ms/step - loss: 0.8176 - accuracy: 0.6989 - val_loss: 0.7662 - val_accuracy: 0.7066
Epoch 18/20
118/118 [==============================] - 29s 249ms/step - loss: 0.8092 - accuracy: 0.7023 - val_loss: 0.7640 - val_accuracy: 0.7133
Epoch 19/20
118/118 [==============================] - 29s 247ms/step - loss: 0.8081 - accuracy: 0.7025 - val_loss: 0.7829 - val_accuracy: 0.6918
Epoch 20/20
118/118 [==============================] - 30s 255ms/step - loss: 0.8022 - accuracy: 0.7049 - val_loss: 0.7619 - val_accuracy: 0.7065
```

Transfert learning using RESNET50



4

Transfert learning using RESNET50

Data augmentation

Model construction

Model Results

Interpretation

▼

- After augmenting data using ‘DataGenerator’ the system doesn’t overfit
- but results didn’t get any better

5 Conclusion: Model with the better performance

Accuracy achieved for Logistic Regression	Accuracy achieved for FFNN	Accuracy achieved for CNN	Accuracy achieved for ResNet50
75%	**87%**	**91%**	**71%**

- We noted that the accuracy achieved by the Logistic regression is not as high the one achieved by FFNN.
- We noted that the accuracy achieved by FFNN model on this dataset is not as high as the CNN model.
- With the ResNet50, we weren’t able to find better results due to the high number of parameters and due to the fact that the model is mostly used for the RBG images.

→ The CNN model was the most performant model out of the four