IMPLEMENTATION

A. Default Neural Network Model (from guide)

The guide presented for this application is a basic clothe image classification neural network model. We are working with 60, 000 images within the Fashion MNIST dataset, this yields 60, 000 labels. Our test set contains 10, 000 of these images, making our labels 10,000. We have already classified these into ten classes: t-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, ankle boot). From these ten classes, our model looks through the dataset and predicts which image falls under which class.

The default model uses one hidden layer of 128 neurons, the input layer having a two-dimensional array of 28 by 28 pixels. And the output layer returns a logits array of length 10. The build of this model can be seen in ***Figure A.1***. This model uses the 'adam' (adaptive moment estimation) algorithm to train the data. Adam is a primarily used neural network algorithm and optimizer training deep learning models.

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Figure A.1. *Implementation of Default Neural Network Model.*

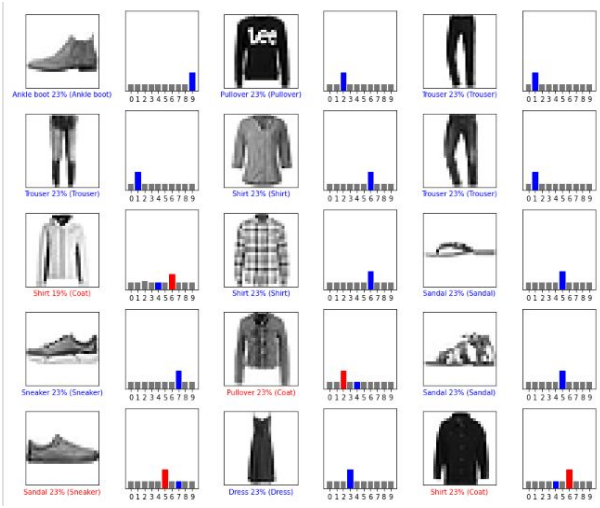

Figure A.2. *Classification of Default Neural Network Model using Adam.*

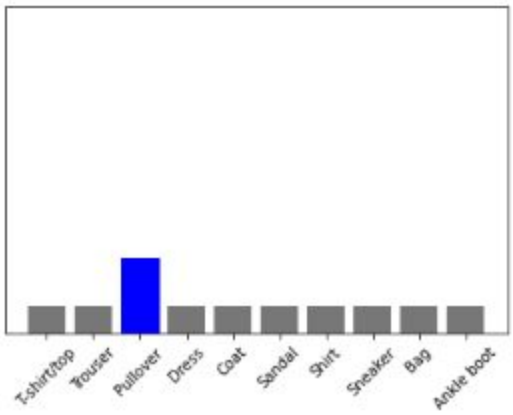

Figure A.2.b. *Classification graph of image[1].*

Within our dataset, image[1] is of a pullover. Our model here is predicting which label this image falls under, and it looks like we are accurate in our prediction. Our model predicts image[1] to be indeed a pullover looking at ***Figure A.2.b***. This is not a surprise seeing as our model's accuracy is approximately 73% (can be checked in Notebook).

Correct predictions in our model are outputted as blue and incorrect as red. Looking at ***Figure A.2***, we can say that the majority of the images are ankle boots, pullovers, trousers, shirts, sandals, sneakers are presented as correct labels. Some are incorrect predictions, such as, 19% shirts, 23% sandals. Let's try using the sgd algorithm and see what our classifications look like.

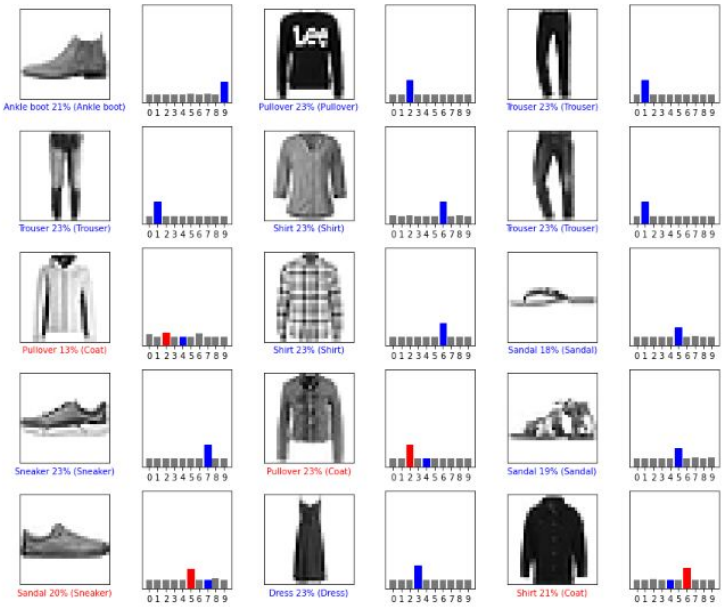

Figure A.3.a. *Classification of Default Neural Network Model using SGD.*

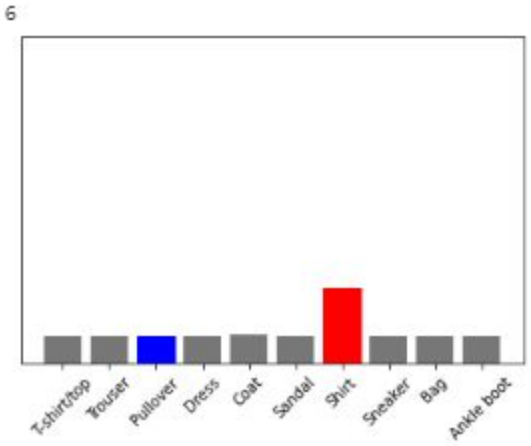

Figure A.3.b. *Classification graph of image[1].*

Our SGD algorithm predicts that image[1] falls under prediction 6, shirt. This is clearly incorrect, this model results in an inaccurate prediction. Although our model displays the shirt label in red as in incorrect prediction, it classifies the image under label 6.

B. Neural Network Model with 100 Neurons

B.1. Five Hidden Layers (20 Neurons Each)

The first case we consider is 100 neurons, we build this model with the help of five hidden layers using 20 neurons per layer. The implementation of the model is displayed in ***Figure B.1.1***.

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(20, activation=tf.nn.relu),
    keras.layers.Dense(20, activation=tf.nn.relu),
    keras.layers.Dense(20, activation=tf.nn.relu),
    keras.layers.Dense(20, activation=tf.nn.relu),
    keras.layers.Dense(20, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Figure B.1.1. *Implementation of the Neural Network Model using Adam.*

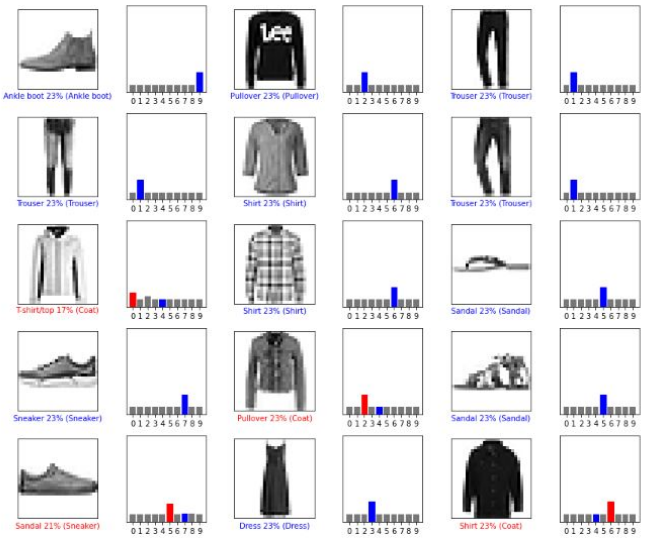

Figure B.1.2.a *Classification of the Neural Network Model using Adam.*

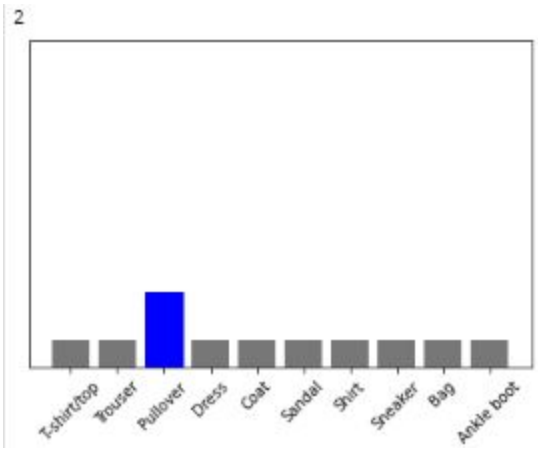

Figure B.1.2.b. *Classification graph of image[1] using Adam.*

```
Test accuracy: 0.7874000072479248
```

We can see that when using five hidden layers with 20 neurons each, our model gives us four incorrect predictions (as seen in ***Figure B.1.2.a***). It is however able to correctly predict image[1] of our dataset (pullover) as seen in ***Figure B.1.2.b***. This model gives us approximately 79% of accuracy.

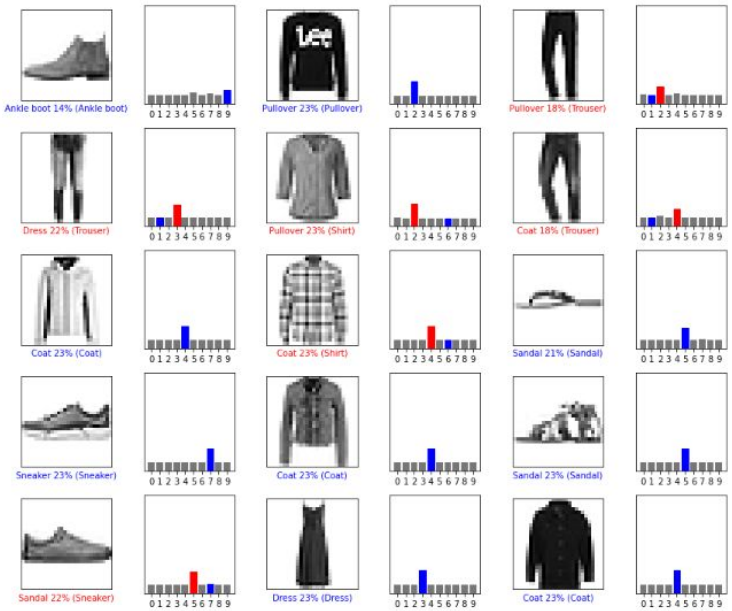

Figure B.1.3.a. *Classification of the Neural Network Model using SGD.*

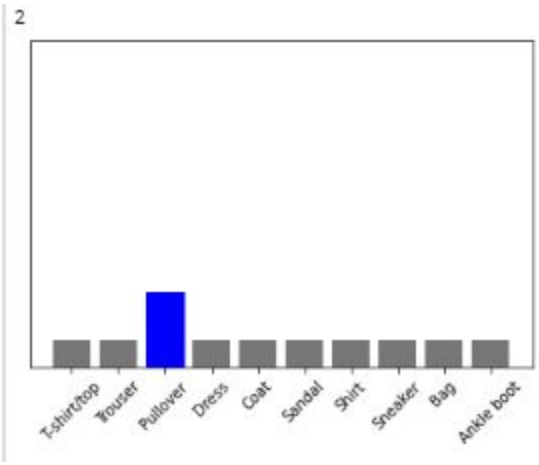

Figure B.1.3.b. *Classification graph of image[1] using SGD.*

```
Test accuracy: 0.6079999804496765
```

Now, when using the SGD algorithm, we can see that here our model gives us five incorrect predictions (as seen in ***Figure B.1.3.a***). Still, we can correctly predict image[1] of our dataset as a pullover as seen in ***Figure B.1.3.b***. This model gives us approximately 61% of accuracy. Here, it's obvious that the Adam algorithm gives us a more accurate prediction.

B.2. Two Hidden Layers (50 Neurons Each)

We consider 100 neurons again, but this time with two hidden layers with 50 neurons each. The implementation of the model is displayed in ***Figure B.1.1***.

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(50, activation=tf.nn.relu),
    keras.layers.Dense(50, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Figure B.2.1. *Implementation of the Neural Network Model using Adam.*

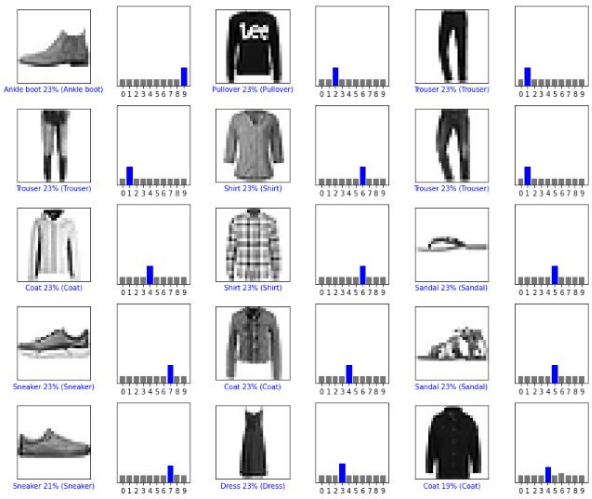

Figure B.2.2.a. *Classification of the Neural Network Model using Adam.*

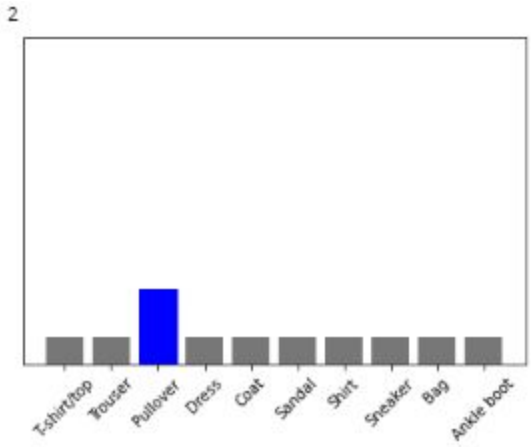

Figure B.2.2.b. *Classification graph of image[1] using Adam.*

```
Test accuracy: 0.8616999983787537
```

For this case using the Adam algorithm, our model yields us a very accurate model. None of our predictions are incorrect, as shown in ***Figure B.1.2.a***. Image[1] is also correctly predicted, as seen in ***Figure B.2.2.b***. This model gives us an accuracy of 86%, so far this model looks to be the most accurate within our cases.

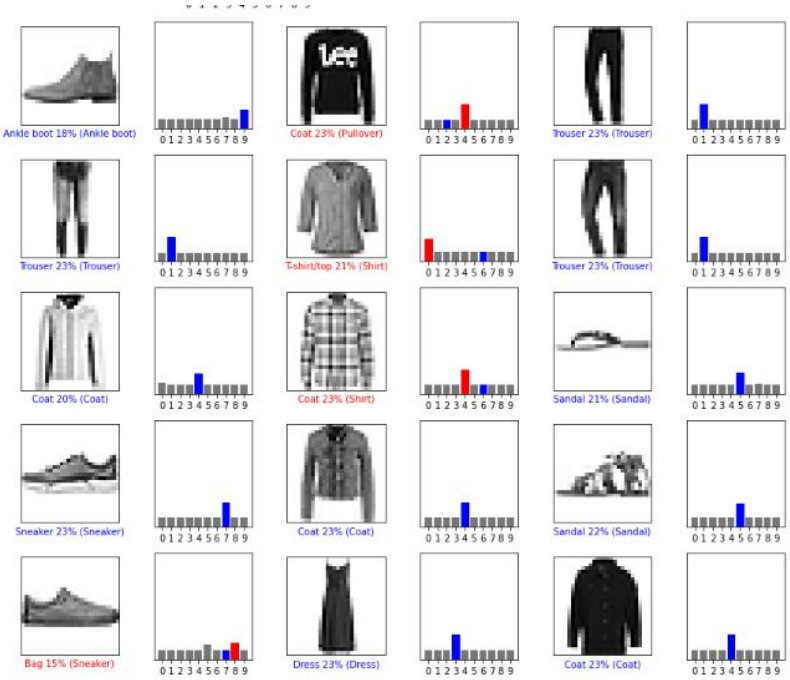

Figure B.2.3.a. *Classification of the Neural Network Model using SGD.*

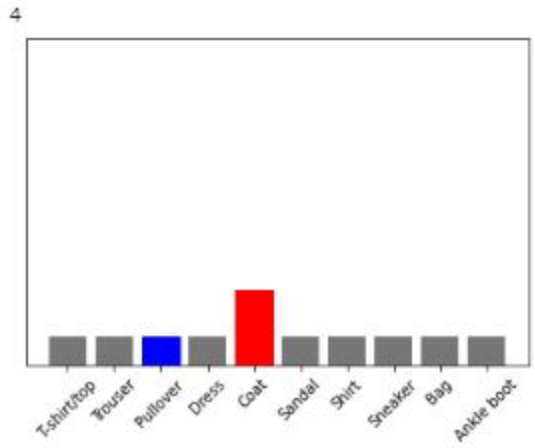

Figure B.2.3.b. *Classification graph of image[1] using SGD.*

```
Test accuracy: 0.7340000027179718
```

Now, looking at the SGD algorithm. We can see that our model gives us four incorrect classifications, including our image[1] classification. This model is not as accurate as the model discussed previously, here we have an accuracy of 73%. We can, once again, say that the Adam algorithm gives us a more accurate representation of clothe image classification.

We used 100 neurons in the previous section (**B.1**), but our models gave us less accurate predictions compared to this section (**B.2**). Considering that, we can say that it is better to have two hidden layers with 50 neurons per layer as opposed to using five hidden layers with 20 neurons each.

C. Neural Network Model with 1000 Neurons

C.1. 5 Hidden Layers (200 Neurons Each)

The next case, we consider 200 neurons, we build this model with the help of five hidden layers using 200 neurons per layer. The implementation of the model is displayed in ***Figure C.1.1***.

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(200, activation=tf.nn.relu),
    keras.layers.Dense(200, activation=tf.nn.relu),
    keras.layers.Dense(200, activation=tf.nn.relu),
    keras.layers.Dense(200, activation=tf.nn.relu),
    keras.layers.Dense(200, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Figure C.1.1. *Implementation of the Neural Network Model using Adam.*

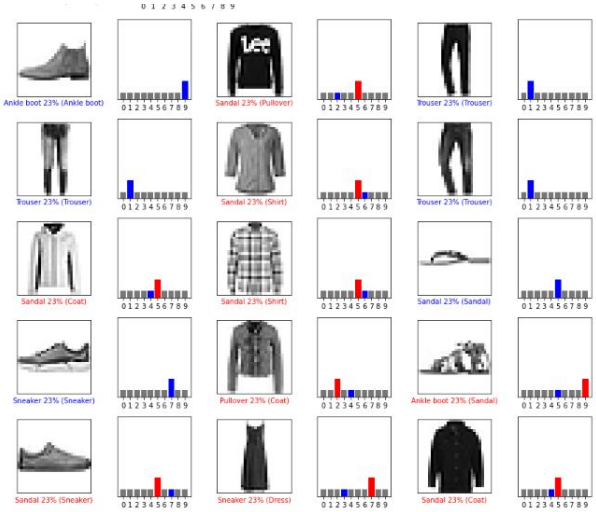

Figure C.1.2.a. *Classification of the Neural Network Model using Adam.*

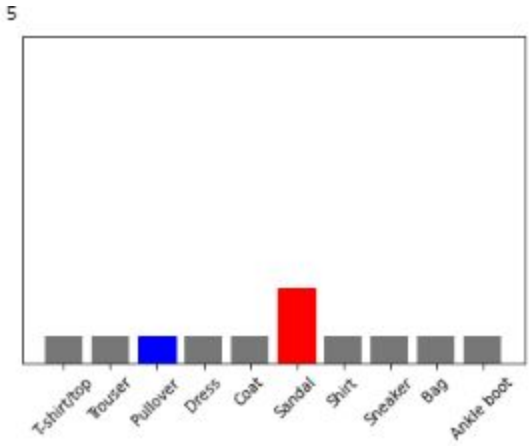

Figure C.1.2.b. *Classification graph of image[1] using Adam.*

```
Test accuracy: 0.10000000149011612
```

This model yields in an accuracy of only 10%. It misclassified the image[1] as sandal and overall, our model gives us 9 incorrect predictions. So far, this is the least accurate model.

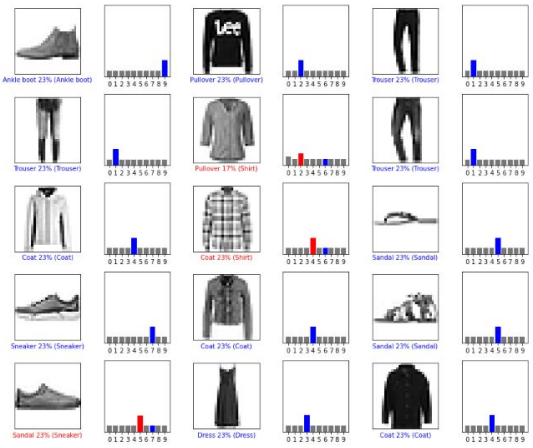

Figure C.1.3.a. *Classification of the Neural Network Model using SGD.*

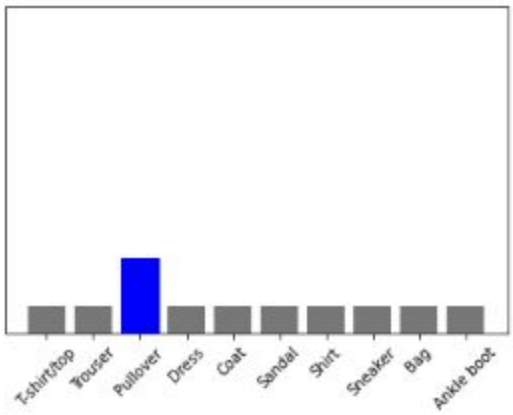

Figure C.1.3.b. *Classification graph of image[1] using SGD.*

```
Test accuracy: 0.8084999918937683
```

Surprisingly, when using the SGD algorithm, we get more accurate predictions. Here, our accuracy rate is approximately 81%, with three incorrect classifications (as seen in ***Figure C.1.3.a***). Our model correctly classifies image[1] as pullover, unlike when using Adam. Here, we can say that the SGD algorithm when using five hidden layers with 200 neurons each is more accurate than when using the Adam algorithm.

C.2. 10 Hidden Layers (100 Neurons Each)

Next, we consider using ten hidden layers with 100 neurons each. The implementation for the built model is displayed in ***Figure C.2.1***.

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(100, activation=tf.nn.relu),
    keras.layers.Dense(100, activation=tf.nn.relu),
    keras.layers.Dense(100, activation=tf.nn.relu),
    keras.layers.Dense(100, activation=tf.nn.relu),
    keras.layers.Dense(100, activation=tf.nn.relu),
    keras.layers.Dense(100, activation=tf.nn.relu),
    keras.layers.Dense(100, activation=tf.nn.relu),
    keras.layers.Dense(100, activation=tf.nn.relu),
    keras.layers.Dense(100, activation=tf.nn.relu),
    keras.layers.Dense(100, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Figure C.2.1. *Implementation of the Neural Network Model using Adam.*

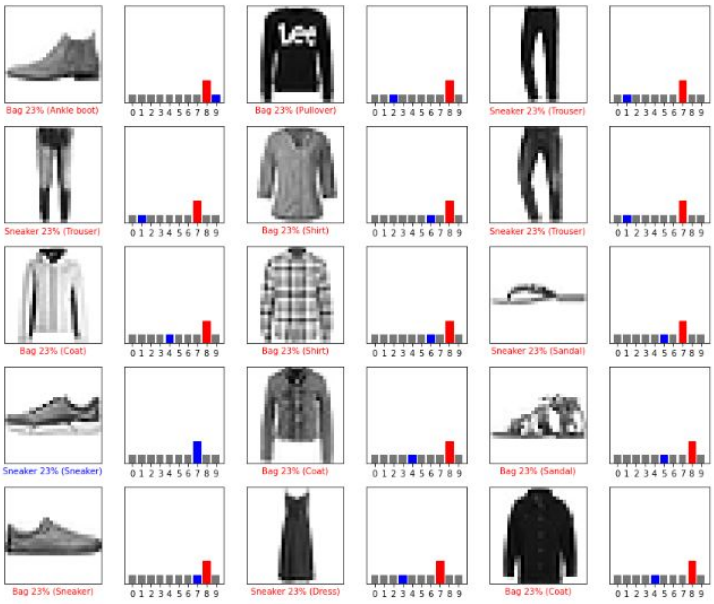

Figure C.2.1.a. *Classification of the Neural Network Model using Adam.*

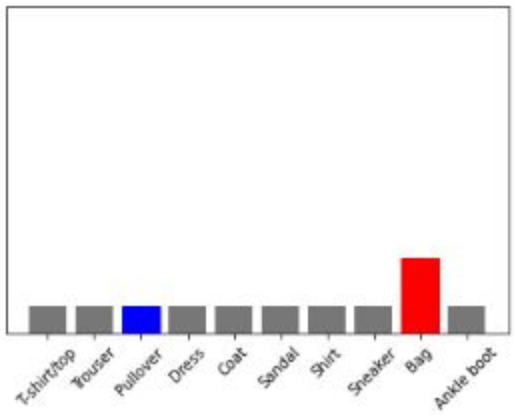

Figure C.2.2.b. *Classification graph of image[1] using Adam.*

```
Test accuracy: 0.10000000149011612
```

Out of 15 predictions, this model gives us 14 incorrect predictions (***Figure C.2.1***), including that of image[1] (***Figure C.2.2.b***). Although it is using the Adam algorithm, this model gives us the least correctly predicted labels yet with an accuracy of 10%. When using Adam for the previous model (using 5 hidden layers with 200 neurons each), our model incorrectly predicted 9 labels. Here, we have 14 incorrect predictions, making this the least accurate model so far.

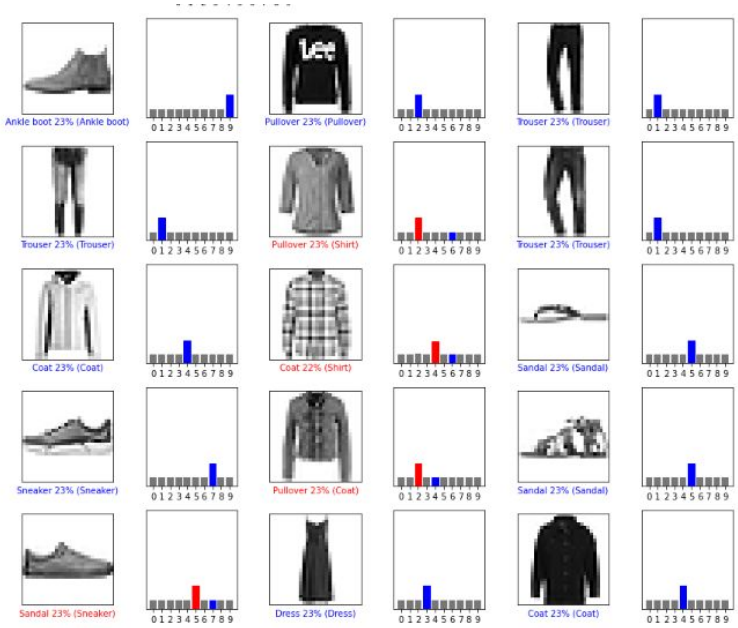

Figure C.2.3.a. *Classification of the Neural Network Model using SGD.*

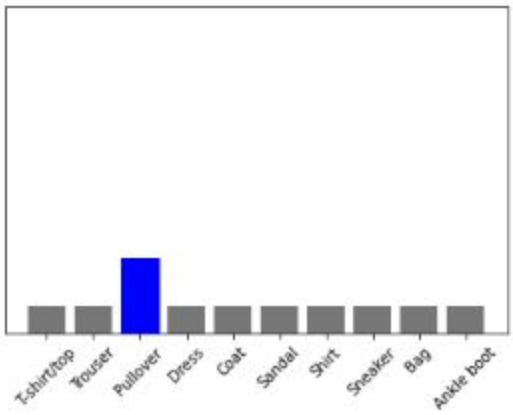

Figure C.2.3.b. *Classification graph of image[1] using SGD.*

```
Test accuracy: 0.781499981880188
```

When using SGD, we get better results. The SGD algorithm gives us an accuracy rate of 78% with only four incorrect predictions. This makes the use of the SGD algorithm more accurate than the use of the Adam algorithm for this case.

D. Neural Network Model with 5000 Neurons

D.1. 5 Hidden Layers (1000 Neurons Each)

We then move onto the next case of 5000 neurons. Here, we first implement a model that uses five hidden layers with 1000 neurons each. The implementation is shown in ***Figure D.1.1***.

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(1000, activation=tf.nn.relu),
    keras.layers.Dense(1000, activation=tf.nn.relu),
    keras.layers.Dense(1000, activation=tf.nn.relu),
    keras.layers.Dense(1000, activation=tf.nn.relu),
    keras.layers.Dense(1000, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Figure D.1.1. *Implementation of the Neural Network Model using Adam.*

Figure D.1.2.a. *Classification of the Neural Network Model using Adam.*

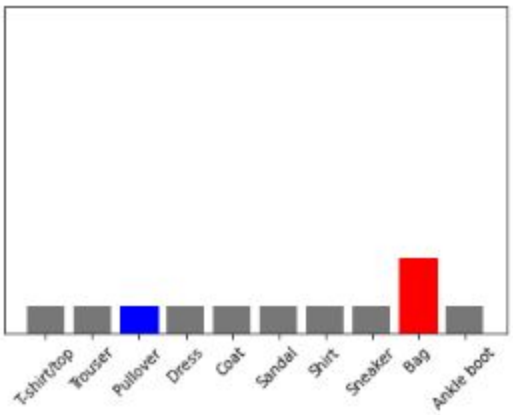

Figure D.1.2.b. *Classification graph of image[1] using Adam.*

```
Test accuracy: 0.10000000149011612
```

When using the Adam algorithm for this case, our model predicts all the labels incorrectly. All 15 predictions are incorrect. This now makes this model the least accurate.

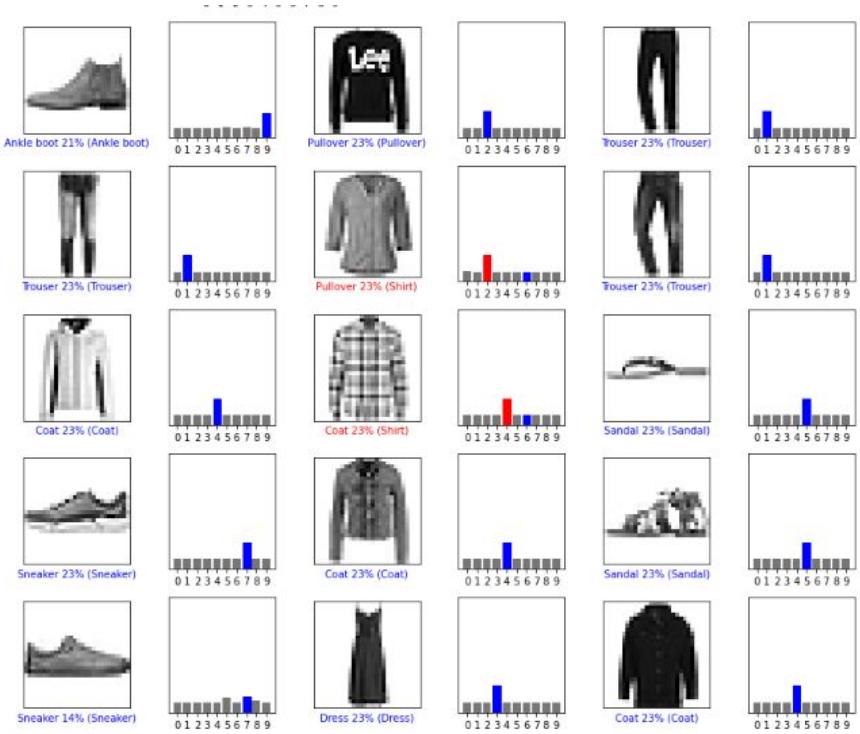

Figure D.1.3.a. *Classification of the Neural Network Model using SGD.*

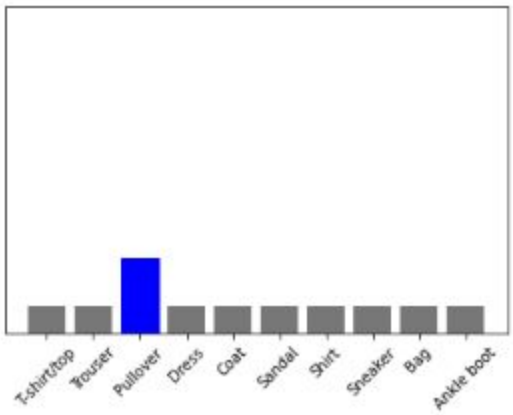

Figure D.1.3.b. *Classification graph of image[1] using SGD.*

```
Test accuracy: 0.8129000067710876
```

When we use the SGD algorithm for this case, our model predicts only two labels incorrectly, unlike when using Adam. Our model gives us an accuracy rate of 81%, which is far more accurate than when we used Adam (with accuracy of 10%). Looking at these two scenarios, we can say the SGD algorithm results in a more accurate model.

D.2. Ten Hidden Layers (500 Neurons Each)

We then consider ten hidden layers with 500 neurons each. This implementation is displayed below in ***Figure D.2.1***.

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(500, activation=tf.nn.relu),
    keras.layers.Dense(500, activation=tf.nn.relu),
    keras.layers.Dense(500, activation=tf.nn.relu),
    keras.layers.Dense(500, activation=tf.nn.relu),
    keras.layers.Dense(500, activation=tf.nn.relu),
    keras.layers.Dense(500, activation=tf.nn.relu),
    keras.layers.Dense(500, activation=tf.nn.relu),
    keras.layers.Dense(500, activation=tf.nn.relu),
    keras.layers.Dense(500, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Figure D.2.1. *Implementation of the Neural Network Model using Adam.*

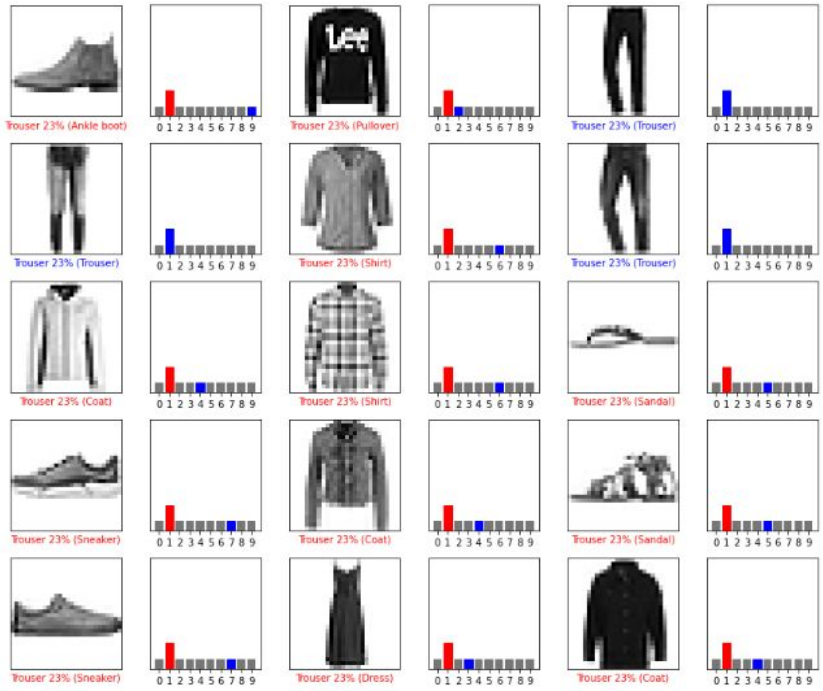

Figure D.2.2.a. *Classification of the Neural Network Model using Adam.*

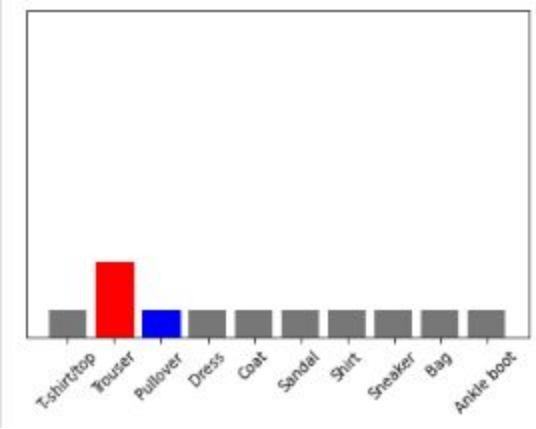

Figure D.2.2.b. *Classification graph of image[1] using Adam.*

```
Test accuracy: 0.10000000149011612
```

Unlike when considering five hidden layers with 1000 neurons each using the Adam algorithm, we now get 12 incorrect predictions. Our **D.2** Adam model is far from accurate but it is more accurate than the Adam algorithm used in **D.1**.

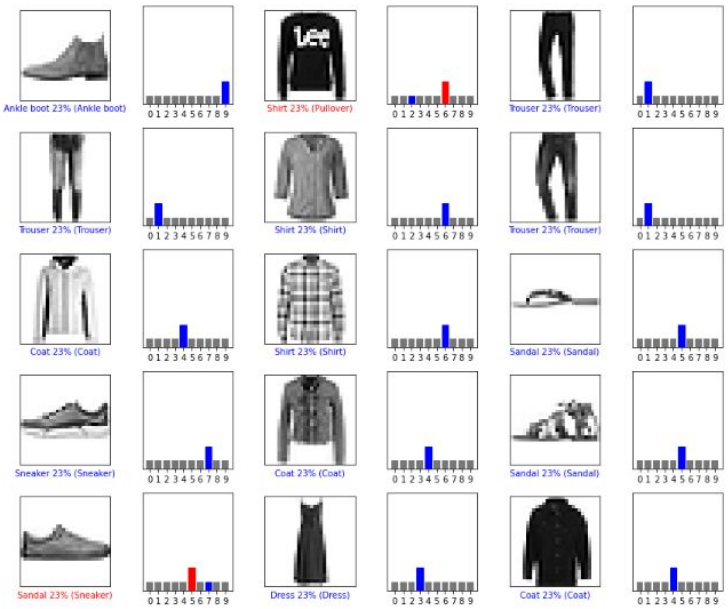

Figure D.2.3.a. *Classification of the Neural Network Model using SGD.*

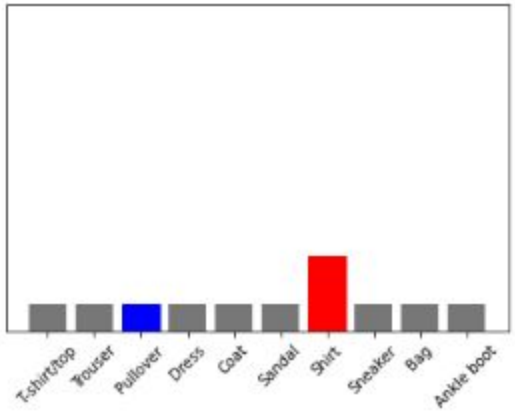

Figure D.2.3.b. *Classification graph of image[1] using SGD.*

```
Test accuracy: 0.7211999893188477
```

When using the SGD algorithm, we now get two incorrect predictions with an accuracy rate of 72%. For this **D.2** case alone, we can conclude that the SGD algorithm gives us a more accurate prediction. But compared to **D.1**, where we used five hidden layers with 1000 neurons each, we can say that the SGD algorithm used in **D.1** is more accurate than when it is used here in **D.2**. This could mean that for the Adam algorithm, less hidden layers with more neurons result in a more accurate model. Whereas, for the SGD algorithm, more hidden layers with less neurons each result in a more accurate model.

E. CNN

Here, we try using the CNN (Convolutional Neural Network) architecture to implement and train our model in order to classify our test set into the 10 labels. We use the same dataset as used before. ***Figure E.1*** displays the model and accuracy using the Adam algorithm. We can see here that this model yields approximately 91%. ***Figure E.2*** shows the model and accuracy using the SGD algorithm. We can see here that this model yields approximately 83%. In our Adam algorithm, our model predicts shirt as dress and coat as shirt, predicts two wrong. Whereas, for SGD, it predicts three incorrect. Hence, we can conclude, based on the accuracy, that the Adam algorithm is more accurate than the SGD.

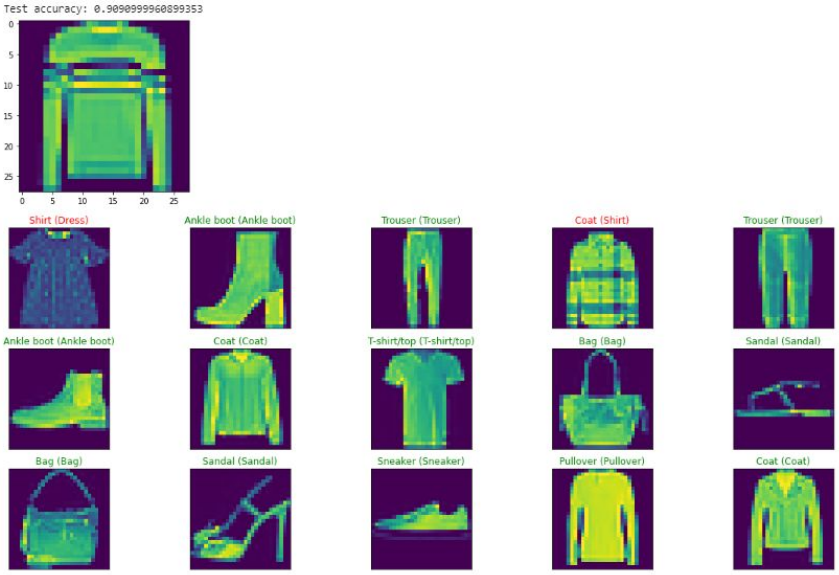

Figure E.1. *CNN model using adam.*

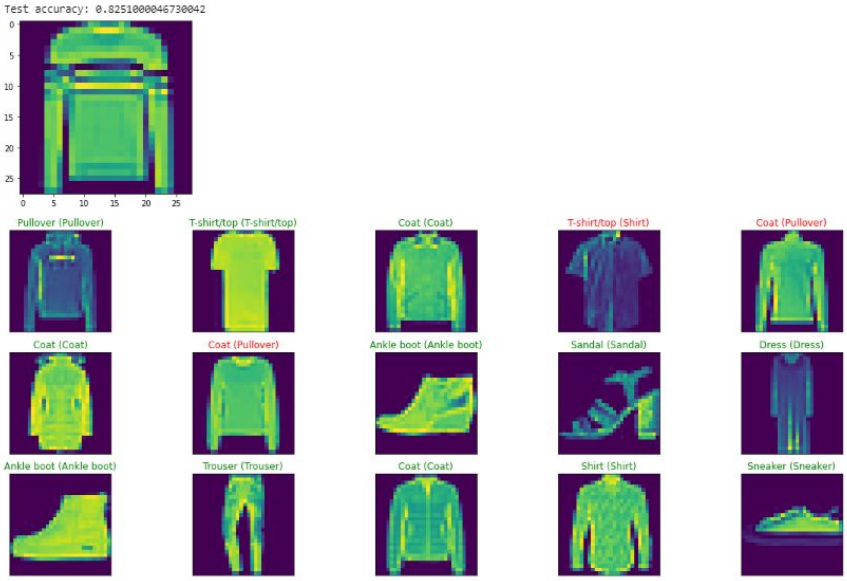

Figure E.2. *CNN model using adam.*

Conclusion

1. **Architecture**

Comparing sections B.1 and B2, we can tell that when we have five hidden layers with 20 neurons each, we have a more inaccurate model as opposed to B.2 (two hidden layers with 50 neurons each). But when we move onto sections C.1 and C.2, it looks like less hidden layers with more neurons is less accurate. But then again, sections D.1 and D.2, we have a similar case as the comparison between B.1 and B.2. Here, less hidden layers with more neurons is more accurate.

2. **Optimizer**

When working with less amount of neurons, it looks like the Adam algorithm is a better optimizer than the SGD (sections A and B). But when we use more neurons, like in sections C and D, our SGD models produce less error than our Adam models. However, using more neurons by default increases error whether we use Adam or SGD. We receive more inaccurate predictions compared to when we use less total neurons. With that in mind, we can also say that when we have less total neurons (100 for example), the Adam algorithm works better. But when we have more total neurons (1000 or 5000), our SGD algorithm works better.

3. **Best Model**

Looking for the best model means looking at the accuracy and correct predictions of our trained models. We have four best models that yield high accuracy, the most accurate model being that of CNN using Adam, this model gave us 91% of accuracy. No other mode that we've trained has given us above 90% accuracy. The second most accurate model is using two hidden layers with 50 neurons each (B.2), giving us 86% of accuracy. This model also predicts the 15 predictions we make correctly, including that of image[1]. The next best model is the CNN model using SGD, giving us an 83% of accuracy. Our final best model, with 81% accuracy, is using five hidden layers with 1000 neurons each (D.1 using SGD). If we continue to increase the amount of total neurons using the SGD algorithm, perhaps we may get a higher accuracy. Including if we decrease total neurons and use the Adam algorithm, we may get a higher accuracy.