Fashion-minst

The code you've shared demonstrates the creation and training of two distinct neural network models for classifying Fashion MNIST images: one is an Artificial Neural Network (ANN), and the other is a Convolutional Neural Network (CNN).

Artificial Neural Network (ANN):

Data Loading:

The code begins by loading the Fashion MNIST dataset, which consists of grayscale images depicting various fashion items.
This dataset is then divided into two sets: a training set (X_train, y_train) and a testing set (X_test, y_test).

Model Architecture:

The ANN is constructed as a Sequential model, where layers are added in a sequential manner.
The initial layer is a Flatten layer that transforms the 28x28 input images into a one-dimensional vector of 784 elements.
Three hidden layers follow with 128, 64, and 32 neurons, respectively, each utilizing the ReLU activation function.
The output layer comprises 10 neurons, corresponding to the number of classes in the Fashion MNIST dataset. It employs the softmax activation function.

Model Compilation:

The model is compiled with the 'adam' optimizer, a popular choice for gradient-based optimization.
'sparse_categorical_crossentropy' is chosen as the loss function since this is a multi-class classification problem.
The evaluation metric used during training is 'accuracy'.

Training:

The ANN model is trained for 10 epochs using the training data, and its performance is assessed using the testing data.
The training history is recorded and stored in the history variable for later analysis.
Convolutional Neural Network (CNN):

Model Architecture:

The CNN model is defined in a Sequential manner as well.
It starts with a Conv2D layer featuring 10 filters, a 3x3 kernel size, and ReLU activation, followed by a MaxPool2D layer with a 2x2 pool size. These layers help the model capture spatial features within the images.
Another set of Conv2D and MaxPool2D layers is added, this time with 15 filters, enabling the model to recognize more intricate patterns.
After flattening the output, three dense layers with 128, 64, and 10 neurons, respectively, are included.
ReLU activation is applied to the hidden layers, while the output layer employs softmax activation.

Model Compilation:

Similar to the ANN, the CNN model uses the 'adam' optimizer.

The loss function remains 'sparse_categorical_crossentropy'.

The evaluation metric chosen for training is 'acc' (accuracy).

Training:

The CNN model is also trained for 10 epochs using the training data, and its performance is evaluated against the testing data.

Training progress is stored in the cnn_history variable for future reference.

Model Comparison and Plotting:

After training both models, the code generates visual plots to facilitate the comparison of their performance.

These plots include training and validation accuracy for both models, as well as training and validation loss for the ANN model.

A final plot is produced to visualize the training accuracy of both the ANN and CNN models for comparison purposes.

In summary, this code serves to contrast the performance of an ANN and a CNN on the Fashion MNIST dataset, with the ANN employing a straightforward feedforward structure and the CNN designed to excel at capturing spatial characteristics in images. Both models utilize the 'adam' optimizer and 'sparse_categorical_crossentropy' loss function, which are commonly employed for similar classification tasks. The resulting training and validation data are depicted graphically to facilitate model performance assessment.