In [1]:
```
# 機械学習 Python Program
# Fashion MNIST dataset の Seqential モデルによる学習と評価並びに検証
```

In [2]:
```
import tensorflow as tf
from tensorflow import keras
```

In [3]:
```
%tensorflow_version 2.x
device_name = tf.test.gpu_device_name()
print('Found GPU at: {}'.format(device_name))
```

```
Found GPU at: /device:GPU:0
```

In [4]:
```
# fashion_mnistの読み込み
fashion_mnist = keras.datasets.fashion_mnist
(X_train_full, y_train_full), (X_test, y_test) = fashion_mnist.load_data()
```

In [5]:
```
# 検証セットの作成＆全てのデータの浮動小数点数への変換（0から1）
X_valid, X_train = X_train_full[:5000] / 255.0, X_train_full[5000:] / 255.0
y_valid, y_train = y_train_full[:5000], y_train_full[5000:]
X_test = X_test / 255.0
```

In [6]:
```
class_names = ["T-shirt/top", "Trouser", "Pullover", "Dress", "Coat",
               "Sandal", "Shirt", "Sneaker", "Bag", "Ankle boot"]
```

In [7]:
```
# シーケンシャルAPIを使ったモデルの作成（層のリスト）
model = keras.models.Sequential([
    keras.layers.Flatten(input_shape=[28, 28]),
    keras.layers.Dense(300, activation="relu"),
    keras.layers.Dense(100, activation="relu"),
    keras.layers.Dense(10, activation="softmax")
])
```

In [8]:
```
# モデルのコンパイル
model.compile(loss="sparse_categorical_crossentropy",
              optimizer="sgd", metrics=["accuracy"])
```

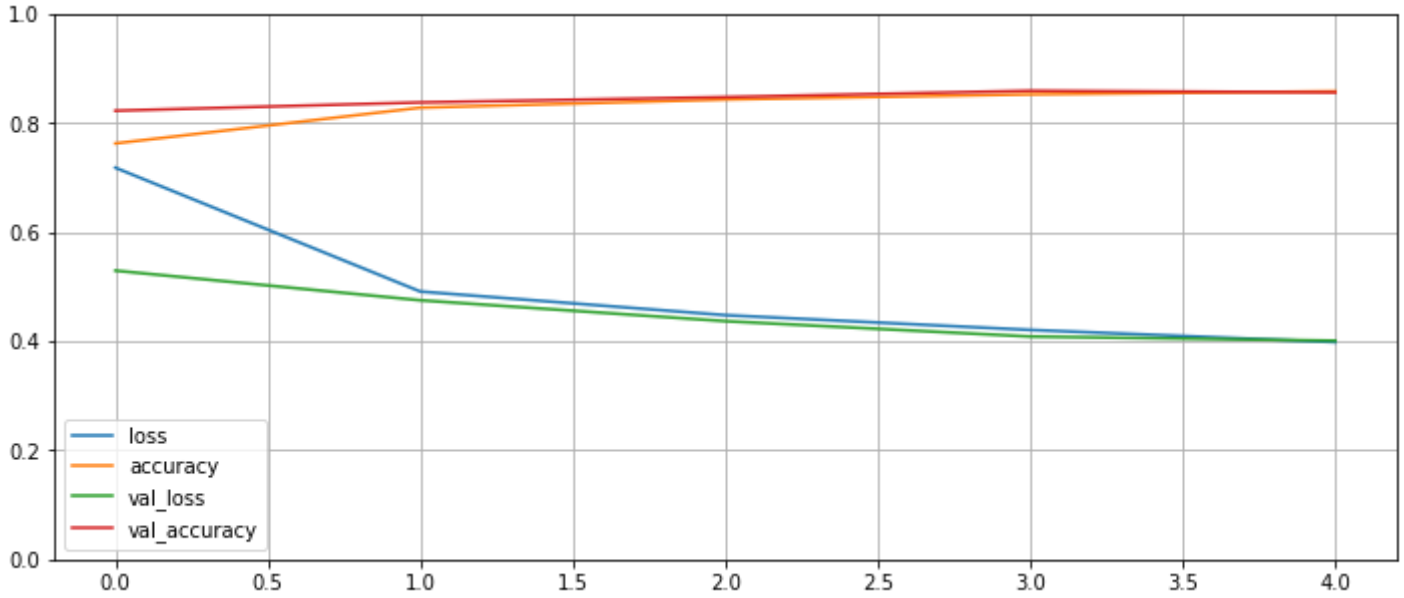
In [9]:
```
# モデルの訓練と評価　本来は、epochs=30
history = model.fit(X_train, y_train, epochs =5, validation_data=(X_valid, y_valid))
```

```
Epoch 1/5
1719/1719 [==============================] - 5s 2ms/step - loss: 0.7183 - accuracy: 0.7628 - val_loss: 0.5296 - val_accuracy: 0.8228
```

```
Epoch 2/5
1719/1719 [==============================] - 4s 2ms/step - loss: 0.4912 - accuracy: 0.8284 - val_loss: 0.4752 - val_accuracy: 0.8380
Epoch 3/5
1719/1719 [==============================] - 4s 2ms/step - loss: 0.4479 - accuracy: 0.8433 - val_loss: 0.4369 - val_accuracy: 0.8478
Epoch 4/5
1719/1719 [==============================] - 4s 2ms/step - loss: 0.4208 - accuracy: 0.8528 - val_loss: 0.4087 - val_accuracy: 0.8594
Epoch 5/5
1719/1719 [==============================] - 4s 2ms/step - loss: 0.3985 - accuracy: 0.8584 - val_loss: 0.4011 - val_accuracy: 0.8564
```

In [10]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [11]:
```
# グラフの描画
pd.DataFrame(history.history).plot(figsize=(12,5))
plt.grid(True)
plt.gca().set_ylim(0, 1) # 縦の範囲を0から1までに
plt.show()
```



In [12]:
```
# 学習の評価
score = model.evaluate(X_test, y_test, verbose=0)
print("Test loss:", score[0])
print("Test acuuracy:", score[1])
```

```
Test loss: 0.4375123977661133
Test acuuracy: 0.8439000248908997
```

In [13]:
```
# モデルを使った予測
num =5
X_new = X_test[:num]
y_pred = model.predict_classes(X_new)
print("予測：")
np.array(class_names)[y_pred]
```

```
予測：
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/sequential.py:455: UserWarning: `model.predict_classes()` is deprecated and will be removed after 2021-01-01. Please use instead:* `np.argmax(model.predict(x), axis=-1)`,   if your model does multi-class classification   (e.g. if it uses a `softmax` last-layer activation).* `(model.predict(x) > 0.5).astype("int32")`,   if your model does binary classification   (e.g. if it uses a `sigmoid` last-layer activation).
  warnings.warn('`model.predict_classes()` is deprecated and '
```

Out[13]:
```
array(['Ankle boot', 'Pullover', 'Trouser', 'Trouser', 'Shirt'],
      dtype='<U11')
```

In [14]:
```
# 正解
y_new = y_test[:num]
print("正解：")
np.array(class_names)[y_new]
```

```
正解：
```

Out[14]:
```
array(['Ankle boot', 'Pullover', 'Trouser', 'Trouser', 'Shirt'],
      dtype='<U11')
```