NeuralNetworks

August 29, 2020

1 Neural Network training

[1]:
```
import tensorflow as tf
from tensorflow import keras
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()

# Preprocess the data (these are NumPy arrays)
x_train = x_train / 255.0
x_test = x_test / 255.0

y_train = y_train.astype("float32")
y_test = y_test.astype("float32")

# Reserve 10,000 samples for validation
x_val = x_train[-10000:]
y_val = y_train[-10000:]
x_train = x_train[:-10000]
y_train = y_train[:-10000]
```

[8]:
```
model = keras.models.Sequential()
model.add(keras.layers.Flatten(input_shape=[28, 28]))
model.add(keras.layers.Dense(300, activation="relu"))
model.add(keras.layers.Dense(300, activation="relu"))
model.add(keras.layers.Dense(10, activation="softmax"))
```

[13]:
```
model.compile(loss="sparse_categorical_crossentropy",
    optimizer="sgd",
    metrics=["accuracy"])
```

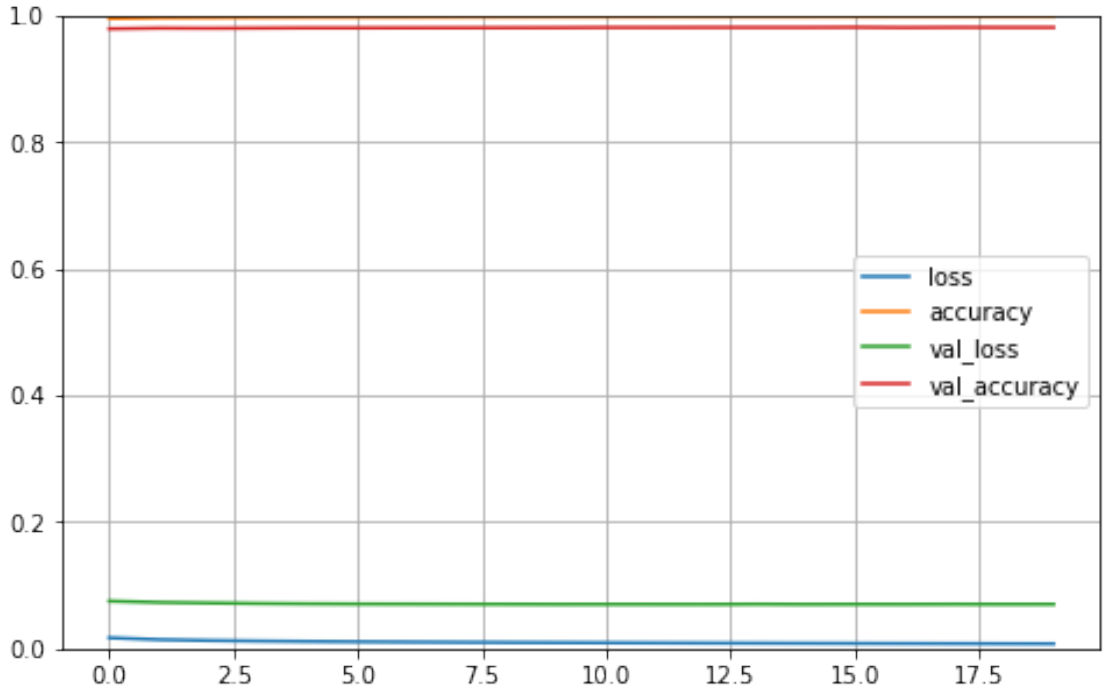
[14]:
```
early_stopping_cb = keras.callbacks.EarlyStopping(patience=5,␣
 ↪restore_best_weights=True)
checkpoint_cb = keras.callbacks.ModelCheckpoint("classifier.h5",␣
 ↪save_best_only=True)
```

```
history = model.fit(x_train, y_train, batch_size=128, epochs=100,␣
↪validation_data=(x_val, y_val), callbacks=[checkpoint_cb, early_stopping_cb])
```

```
Epoch 1/100
391/391 [==============================] - 1s 3ms/step - loss: 0.0171 -
accuracy: 0.9952 - val_loss: 0.0751 - val_accuracy: 0.9792
Epoch 2/100
391/391 [==============================] - 1s 2ms/step - loss: 0.0142 -
accuracy: 0.9965 - val_loss: 0.0731 - val_accuracy: 0.9802
Epoch 3/100
391/391 [==============================] - 1s 2ms/step - loss: 0.0128 -
accuracy: 0.9969 - val_loss: 0.0721 - val_accuracy: 0.9800
Epoch 4/100
391/391 [==============================] - 1s 2ms/step - loss: 0.0120 -
accuracy: 0.9973 - val_loss: 0.0713 - val_accuracy: 0.9803
Epoch 5/100
391/391 [==============================] - 1s 3ms/step - loss: 0.0113 -
accuracy: 0.9976 - val_loss: 0.0708 - val_accuracy: 0.9806
Epoch 6/100
391/391 [==============================] - 1s 2ms/step - loss: 0.0109 -
accuracy: 0.9977 - val_loss: 0.0705 - val_accuracy: 0.9806
Epoch 7/100
391/391 [==============================] - 1s 2ms/step - loss: 0.0104 -
accuracy: 0.9977 - val_loss: 0.0704 - val_accuracy: 0.9807
Epoch 8/100
391/391 [==============================] - 1s 3ms/step - loss: 0.0101 -
accuracy: 0.9980 - val_loss: 0.0703 - val_accuracy: 0.9808
Epoch 9/100
391/391 [==============================] - 1s 3ms/step - loss: 0.0098 -
accuracy: 0.9981 - val_loss: 0.0702 - val_accuracy: 0.9809
Epoch 10/100
391/391 [==============================] - 1s 3ms/step - loss: 0.0095 -
accuracy: 0.9983 - val_loss: 0.0700 - val_accuracy: 0.9810
Epoch 11/100
391/391 [==============================] - 1s 3ms/step - loss: 0.0093 -
accuracy: 0.9982 - val_loss: 0.0700 - val_accuracy: 0.9813
Epoch 12/100
391/391 [==============================] - 1s 2ms/step - loss: 0.0090 -
accuracy: 0.9984 - val_loss: 0.0700 - val_accuracy: 0.9813
Epoch 13/100
391/391 [==============================] - 1s 3ms/step - loss: 0.0088 -
accuracy: 0.9984 - val_loss: 0.0699 - val_accuracy: 0.9814
Epoch 14/100
391/391 [==============================] - 1s 2ms/step - loss: 0.0086 -
accuracy: 0.9984 - val_loss: 0.0703 - val_accuracy: 0.9814
Epoch 15/100
391/391 [==============================] - 1s 2ms/step - loss: 0.0085 -
```

```
accuracy: 0.9984 - val_loss: 0.0699 - val_accuracy: 0.9814
Epoch 16/100
391/391 [==============================] - 1s 3ms/step - loss: 0.0083 -
accuracy: 0.9985 - val_loss: 0.0701 - val_accuracy: 0.9816
Epoch 17/100
391/391 [==============================] - 1s 2ms/step - loss: 0.0081 -
accuracy: 0.9986 - val_loss: 0.0700 - val_accuracy: 0.9812
Epoch 18/100
391/391 [==============================] - 1s 3ms/step - loss: 0.0080 -
accuracy: 0.9985 - val_loss: 0.0702 - val_accuracy: 0.9815
Epoch 19/100
391/391 [==============================] - 1s 3ms/step - loss: 0.0078 -
accuracy: 0.9986 - val_loss: 0.0701 - val_accuracy: 0.9814
Epoch 20/100
391/391 [==============================] - 1s 3ms/step - loss: 0.0077 -
accuracy: 0.9986 - val_loss: 0.0701 - val_accuracy: 0.9814
```

[15]:
```
pd.DataFrame(history.history).plot(figsize=(8, 5))
plt.grid(True)
plt.gca().set_ylim(0, 1) # set the vertical range to [0-1]
plt.show()
```



[16]:
```
model.evaluate(x_test, y_test)
```

```
313/313 [==============================] - 0s 855us/step - loss: 0.0567 -
```

```
accuracy: 0.9833
```

[16]: [0.056724756956100464, 0.983299970626831]