Multi_Layer_Perceptron

April 8, 2020

[0]:
```
import numpy as np
import tensorflow as tf
from tensorflow.keras.layers import Input, Flatten, Dense
from tensorflow.keras.models import Model
from tensorflow.keras.datasets import mnist
from matplotlib import pyplot as plt
```

[0]:
```
(X_train, y_train), (X_test, y_test) = mnist.load_data()

X_train = X_train / 255.
X_test = X_test / 255.
```

[25]:
```
input_img = Input(shape=(28, 28))
x = Flatten()(input_img)
x = Dense(50, activation="relu")(x)
classification_layer = Dense(10)(x)

model = Model(input_img, classification_layer)
model.compile(optimizer=tf.keras.optimizers.Adam(lr=0.01),
              loss=tf.keras.losses.
 ↪SparseCategoricalCrossentropy(from_logits=True),
              metrics=["accuracy"])
model.summary()

model.fit(X_train, y_train,
          epochs=10,
          batch_size=128,
          shuffle=True,
          validation_data=(X_test, y_test))
```

```
Model: "model_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_10 (InputLayer)        [(None, 28, 28)]          0
_________________________________________________________________
flatten_8 (Flatten)          (None, 784)               0
_________________________________________________________________
```

```
dense_14 (Dense)             (None, 50)                39250
_________________________________________________________________
dense_15 (Dense)             (None, 10)                510
=================================================================
Total params: 39,760
Trainable params: 39,760
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
469/469 [==============================] - 1s 3ms/step - loss: 0.2623 -
accuracy: 0.9216 - val_loss: 0.1688 - val_accuracy: 0.9490
Epoch 2/10
469/469 [==============================] - 1s 3ms/step - loss: 0.1392 -
accuracy: 0.9583 - val_loss: 0.1367 - val_accuracy: 0.9586
Epoch 3/10
469/469 [==============================] - 1s 3ms/step - loss: 0.1114 -
accuracy: 0.9655 - val_loss: 0.1449 - val_accuracy: 0.9589
Epoch 4/10
469/469 [==============================] - 1s 3ms/step - loss: 0.1016 -
accuracy: 0.9697 - val_loss: 0.1468 - val_accuracy: 0.9581
Epoch 5/10
469/469 [==============================] - 1s 3ms/step - loss: 0.0863 -
accuracy: 0.9738 - val_loss: 0.1397 - val_accuracy: 0.9640
Epoch 6/10
469/469 [==============================] - 1s 3ms/step - loss: 0.0823 -
accuracy: 0.9746 - val_loss: 0.1618 - val_accuracy: 0.9565
Epoch 7/10
469/469 [==============================] - 1s 3ms/step - loss: 0.0758 -
accuracy: 0.9768 - val_loss: 0.1428 - val_accuracy: 0.9653
Epoch 8/10
469/469 [==============================] - 1s 3ms/step - loss: 0.0740 -
accuracy: 0.9772 - val_loss: 0.1493 - val_accuracy: 0.9660
Epoch 9/10
469/469 [==============================] - 1s 3ms/step - loss: 0.0689 -
accuracy: 0.9785 - val_loss: 0.1400 - val_accuracy: 0.9685
Epoch 10/10
469/469 [==============================] - 1s 3ms/step - loss: 0.0632 -
accuracy: 0.9803 - val_loss: 0.1485 - val_accuracy: 0.9674
```

[25]:
```
<tensorflow.python.keras.callbacks.History at 0x7f001f298780>
```

[26]:
```
input_img = Input(shape=(28, 28))
x = Flatten()(input_img)
x = Dense(150, activation="relu")(x)
classification_layer = Dense(10)(x)

model = Model(input_img, classification_layer)
```

```
model.compile(optimizer=tf.keras.optimizers.Adam(lr=0.01),
              loss=tf.keras.losses.
 ↪SparseCategoricalCrossentropy(from_logits=True),
              metrics=["accuracy"])
model.summary()

model.fit(X_train, y_train,
          epochs=10,
          batch_size=128,
          shuffle=True,
          validation_data=(X_test, y_test))
```

```
Model: "model_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_11 (InputLayer)        [(None, 28, 28)]          0
_________________________________________________________________
flatten_9 (Flatten)          (None, 784)               0
_________________________________________________________________
dense_16 (Dense)             (None, 150)               117750
_________________________________________________________________
dense_17 (Dense)             (None, 10)                1510
=================================================================
Total params: 119,260
Trainable params: 119,260
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
469/469 [==============================] - 2s 4ms/step - loss: 0.2224 -
accuracy: 0.9329 - val_loss: 0.1084 - val_accuracy: 0.9671
Epoch 2/10
469/469 [==============================] - 2s 4ms/step - loss: 0.1077 -
accuracy: 0.9672 - val_loss: 0.1129 - val_accuracy: 0.9653
Epoch 3/10
469/469 [==============================] - 2s 4ms/step - loss: 0.0851 -
accuracy: 0.9737 - val_loss: 0.1306 - val_accuracy: 0.9633
Epoch 4/10
469/469 [==============================] - 2s 4ms/step - loss: 0.0800 -
accuracy: 0.9750 - val_loss: 0.1037 - val_accuracy: 0.9716
Epoch 5/10
469/469 [==============================] - 2s 4ms/step - loss: 0.0647 -
accuracy: 0.9801 - val_loss: 0.1398 - val_accuracy: 0.9668
Epoch 6/10
469/469 [==============================] - 2s 4ms/step - loss: 0.0643 -
accuracy: 0.9808 - val_loss: 0.1442 - val_accuracy: 0.9684
Epoch 7/10
```

```
469/469 [==============================] - 2s 4ms/step - loss: 0.0634 -
accuracy: 0.9814 - val_loss: 0.1412 - val_accuracy: 0.9683
Epoch 8/10
469/469 [==============================] - 2s 4ms/step - loss: 0.0615 -
accuracy: 0.9824 - val_loss: 0.1272 - val_accuracy: 0.9717
Epoch 9/10
469/469 [==============================] - 2s 4ms/step - loss: 0.0477 -
accuracy: 0.9862 - val_loss: 0.1495 - val_accuracy: 0.9723
Epoch 10/10
469/469 [==============================] - 2s 4ms/step - loss: 0.0519 -
accuracy: 0.9856 - val_loss: 0.1493 - val_accuracy: 0.9702
```

[26]:
```
<tensorflow.python.keras.callbacks.History at 0x7f001f35ef60>
```

[35]:
```
input_img = Input(shape=(28, 28))
x = Flatten()(input_img)
x = Dense(50, activation="relu", kernel_regularizer=tf.keras.regularizers.
 ↪l2(l=0.001))(x)
classification_layer = Dense(10)(x)

model = Model(input_img, classification_layer)
model.compile(optimizer=tf.keras.optimizers.Adam(lr=0.01),
              loss=tf.keras.losses.
 ↪SparseCategoricalCrossentropy(from_logits=True),
              metrics=["accuracy"])
model.summary()

model.fit(X_train, y_train,
          epochs=10,
          batch_size=128,
          shuffle=True,
          validation_data=(X_test, y_test))
```

```
Model: "model_17"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_20 (InputLayer)        [(None, 28, 28)]          0
_________________________________________________________________
flatten_18 (Flatten)         (None, 784)               0
_________________________________________________________________
dense_34 (Dense)             (None, 50)                39250
_________________________________________________________________
dense_35 (Dense)             (None, 10)                510
=================================================================
Total params: 39,760
Trainable params: 39,760
Non-trainable params: 0
```

```
----------------------------------------------------------------
Epoch 1/10
469/469 [==============================] - 1s 3ms/step - loss: 0.4018 -
accuracy: 0.9128 - val_loss: 0.3123 - val_accuracy: 0.9390
Epoch 2/10
469/469 [==============================] - 1s 3ms/step - loss: 0.3264 -
accuracy: 0.9380 - val_loss: 0.3508 - val_accuracy: 0.9288
Epoch 3/10
469/469 [==============================] - 1s 3ms/step - loss: 0.3132 -
accuracy: 0.9417 - val_loss: 0.3090 - val_accuracy: 0.9392
Epoch 4/10
469/469 [==============================] - 1s 3ms/step - loss: 0.3144 -
accuracy: 0.9430 - val_loss: 0.2880 - val_accuracy: 0.9468
Epoch 5/10
469/469 [==============================] - 1s 3ms/step - loss: 0.3057 -
accuracy: 0.9450 - val_loss: 0.3624 - val_accuracy: 0.9268
Epoch 6/10
469/469 [==============================] - 1s 3ms/step - loss: 0.3090 -
accuracy: 0.9431 - val_loss: 0.2758 - val_accuracy: 0.9542
Epoch 7/10
469/469 [==============================] - 1s 3ms/step - loss: 0.3030 -
accuracy: 0.9452 - val_loss: 0.3045 - val_accuracy: 0.9436
Epoch 8/10
469/469 [==============================] - 1s 3ms/step - loss: 0.2968 -
accuracy: 0.9463 - val_loss: 0.2992 - val_accuracy: 0.9449
Epoch 9/10
469/469 [==============================] - 1s 3ms/step - loss: 0.2971 -
accuracy: 0.9452 - val_loss: 0.3073 - val_accuracy: 0.9428
Epoch 10/10
469/469 [==============================] - 1s 3ms/step - loss: 0.2899 -
accuracy: 0.9469 - val_loss: 0.2822 - val_accuracy: 0.9498
```

[35]:
```
<tensorflow.python.keras.callbacks.History at 0x7f0017f90390>
```

[36]:
```
input_img = Input(shape=(28, 28))
x = Flatten()(input_img)
x = Dense(250, activation="relu", kernel_regularizer=tf.keras.regularizers.
 ↪l2(l=0.001))(x)
classification_layer = Dense(10)(x)

model = Model(input_img, classification_layer)
model.compile(optimizer=tf.keras.optimizers.Adam(lr=0.01),
              loss=tf.keras.losses.
 ↪SparseCategoricalCrossentropy(from_logits=True),
              metrics=["accuracy"])
model.summary()
```

```
model.fit(X_train, y_train,
          epochs=10,
          batch_size=128,
          shuffle=True,
          validation_data=(X_test, y_test))
```

```
Model: "model_18"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_21 (InputLayer)        [(None, 28, 28)]          0
_________________________________________________________________
flatten_19 (Flatten)         (None, 784)               0
_________________________________________________________________
dense_36 (Dense)             (None, 250)               196250
_________________________________________________________________
dense_37 (Dense)             (None, 10)                2510
=================================================================
Total params: 198,760
Trainable params: 198,760
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
469/469 [==============================] - 2s 5ms/step - loss: 0.4713 -
accuracy: 0.9169 - val_loss: 0.4151 - val_accuracy: 0.9310
Epoch 2/10
469/469 [==============================] - 2s 5ms/step - loss: 0.3876 -
accuracy: 0.9366 - val_loss: 0.3628 - val_accuracy: 0.9380
Epoch 3/10
469/469 [==============================] - 2s 5ms/step - loss: 0.3545 -
accuracy: 0.9412 - val_loss: 0.3443 - val_accuracy: 0.9449
Epoch 4/10
469/469 [==============================] - 2s 5ms/step - loss: 0.3344 -
accuracy: 0.9439 - val_loss: 0.3107 - val_accuracy: 0.9479
Epoch 5/10
469/469 [==============================] - 2s 5ms/step - loss: 0.3278 -
accuracy: 0.9437 - val_loss: 0.3043 - val_accuracy: 0.9489
Epoch 6/10
469/469 [==============================] - 2s 5ms/step - loss: 0.3036 -
accuracy: 0.9481 - val_loss: 0.3055 - val_accuracy: 0.9406
Epoch 7/10
469/469 [==============================] - 2s 5ms/step - loss: 0.3078 -
accuracy: 0.9466 - val_loss: 0.3018 - val_accuracy: 0.9443
Epoch 8/10
469/469 [==============================] - 2s 5ms/step - loss: 0.3102 -
accuracy: 0.9456 - val_loss: 0.2688 - val_accuracy: 0.9556
Epoch 9/10
```

```
469/469 [==============================] - 2s 5ms/step - loss: 0.2951 -
accuracy: 0.9484 - val_loss: 0.3266 - val_accuracy: 0.9393
Epoch 10/10
469/469 [==============================] - 2s 5ms/step - loss: 0.3018 -
accuracy: 0.9465 - val_loss: 0.2969 - val_accuracy: 0.9453
```

[36]: `<tensorflow.python.keras.callbacks.History at 0x7f00183259b0>`

	Classification accuracy	
	Training	Testing
50HLN+no regularization	0.9803	0.9674
50HLN+L2 regularization	0.9469	0.9498
250HLN+no regularization	0.9856	0.9702
250HLN+L2 regularization	0.9465	0.9453

The number of epochs is set to 10 and the learning rate to 0.01 for all configurations.

The table shows that the regularization term has a negative impact on the training accuracy when compared to the MLPs with no regularization term. However, the testing error is now approximately the same as the training error, meaning that the model extrapolates well to unseen data. This is opposed to the case with no regularization, where the testing accuracy is lower than the training accuracy.

[0]: