TensorFlow Worknotes

Fadjar Fathurrahman

1 Simple neuron

Define the model

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

model = Sequential()
model.add( Dense(1, batch_input_shape=(None,2), activation="sigmoid") )
```

```
from tensorflow.keras import optimizers
sgd = optimizers.SGD(lr=0.15)
```

Compile the model, using `binary_crossentropy`.

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

Print model summary

```
print(model.summary())
```

Train the model

```
history = model.fit(
    X, Y,epochs=400,
    batch_size=128,
    verbose=1
)
```

Example of prediction:

```
model.predict( np.array([[1.0, 1.0]]) )
```

Note that the input should be an array of size `(1,2)`. The first dimension can be of any size instead of one.

2 Using hidden layer

Second layer is added with size 8, output layer with softmax activation.

```
model = Sequential()
model.add( Dense(8, batch_input_shape=(None,2), activation="sigmoid") )
model.add( Dense(2, activation='softmax'))
```

Second (output) layer has two nodes corresponding to two classes.

We need to convert the target value to categorical:

```
# Transform Y=0 to (1,0) and Y=1 to (0,1)
Y_c = to_categorical(Y, 2)
```

Compile the model, using `categorical_crossentropy` loss function:

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

Training:

```
history = model.fit(
    X, Y_c,epochs=400,
    batch_size=128,
    verbose=1
)
```