

FliK Modul 2020

Data Handling and Keras

Steffen Seitz, Marvin Arnold & Markus Fritzsche

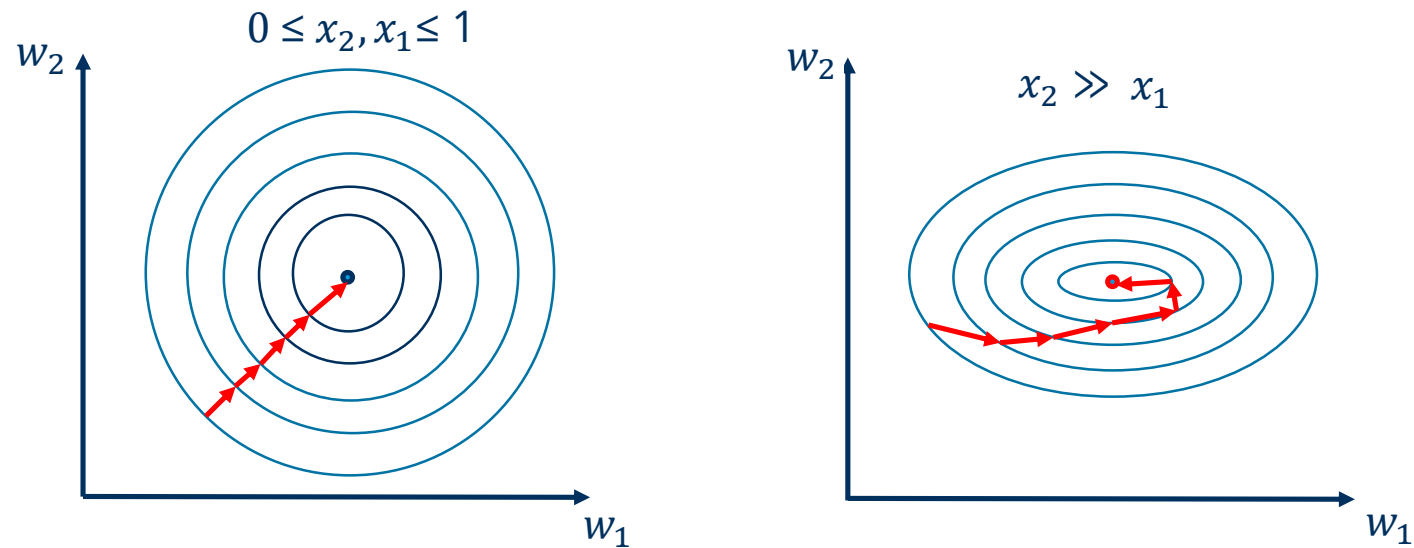
Prof. Ronald Tetzlaff

Dresden, 19-23.10.

Data Pre-Adjustments

Normalization

Data is normalized, before given to the network.

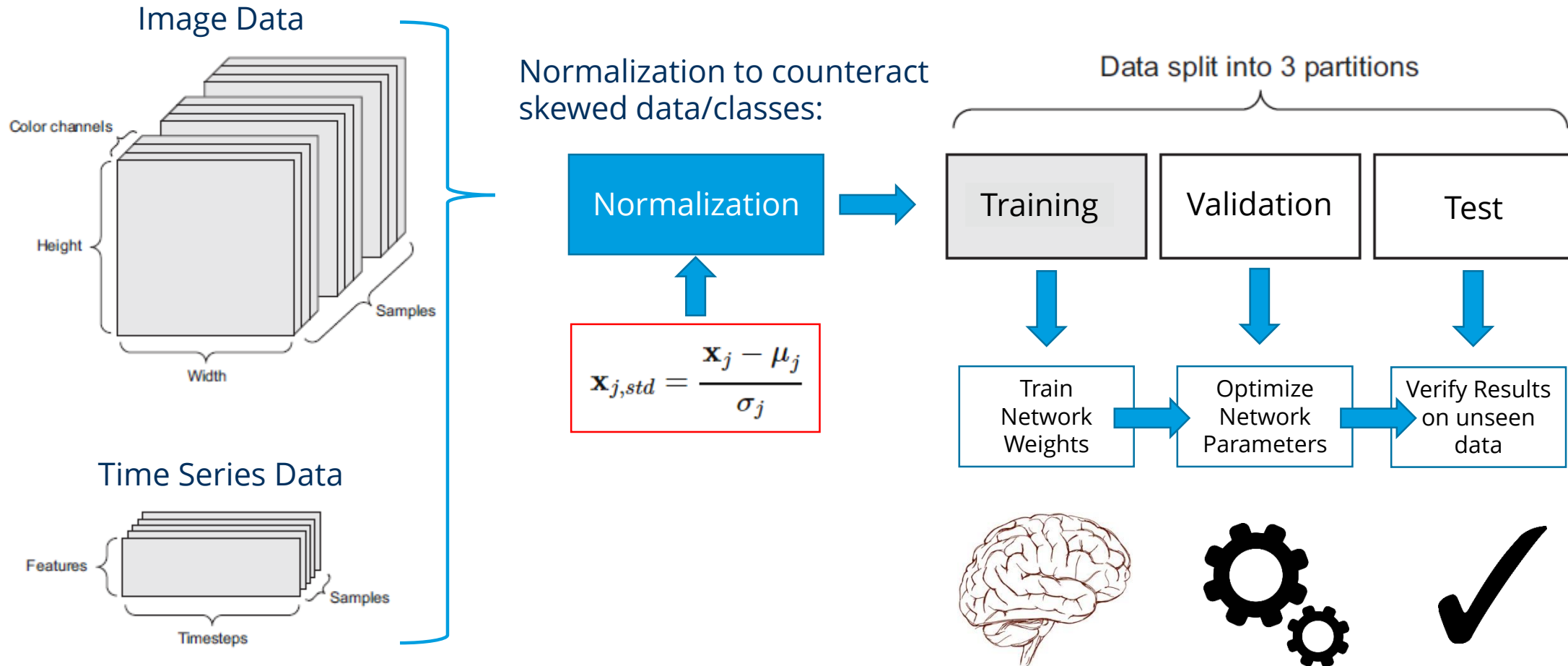


For **skewed classes**, learning will be slower!

This is why you need to scale the input values to have unit variance and zero mean:

Data Pre-Adjustments

Train Test Validation splitting



Label Pre-Adjustments

One-Hot Encoding

index	label
0	airplane (0)
1	automobile (1)
2	bird (2)
3	cat (3)
4	deer (4)
5	dog (5)
6	frog (6)
7	horse (7)
8	ship (8)
9	truck (9)
...	...
...	...

original label data



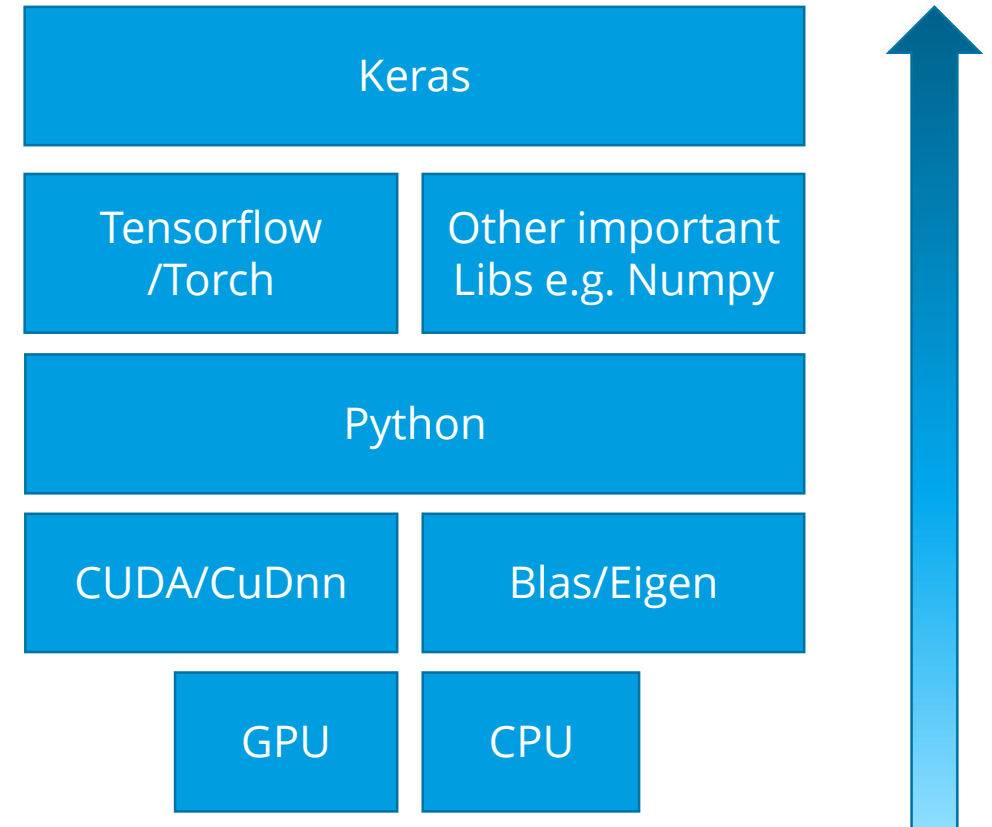
label	index											
	0	1	2	3	4	5	6	7	8	9
airplane	1	0	0	0	0	0	0	0	0	0
automobile	0	1	0	0	0	0	0	0	0	0
bird	0	0	1	0	0	0	0	0	0	0
cat	0	0	0	1	0	0	0	0	0	0
deer	0	0	0	0	1	0	0	0	0	0
dog	0	0	0	0	0	1	0	0	0	0
frog	0	0	0	0	0	0	1	0	0	0
horse	0	0	0	0	0	0	0	1	0	0
ship	0	0	0	0	0	0	0	0	1	0
truck	0	0	0	0	0	0	0	0	0	1

one-hot-encoded label data

The Keras Software & Hardware Stack

Keras is a „top level“ python class inheriting from other important libraries written in python.

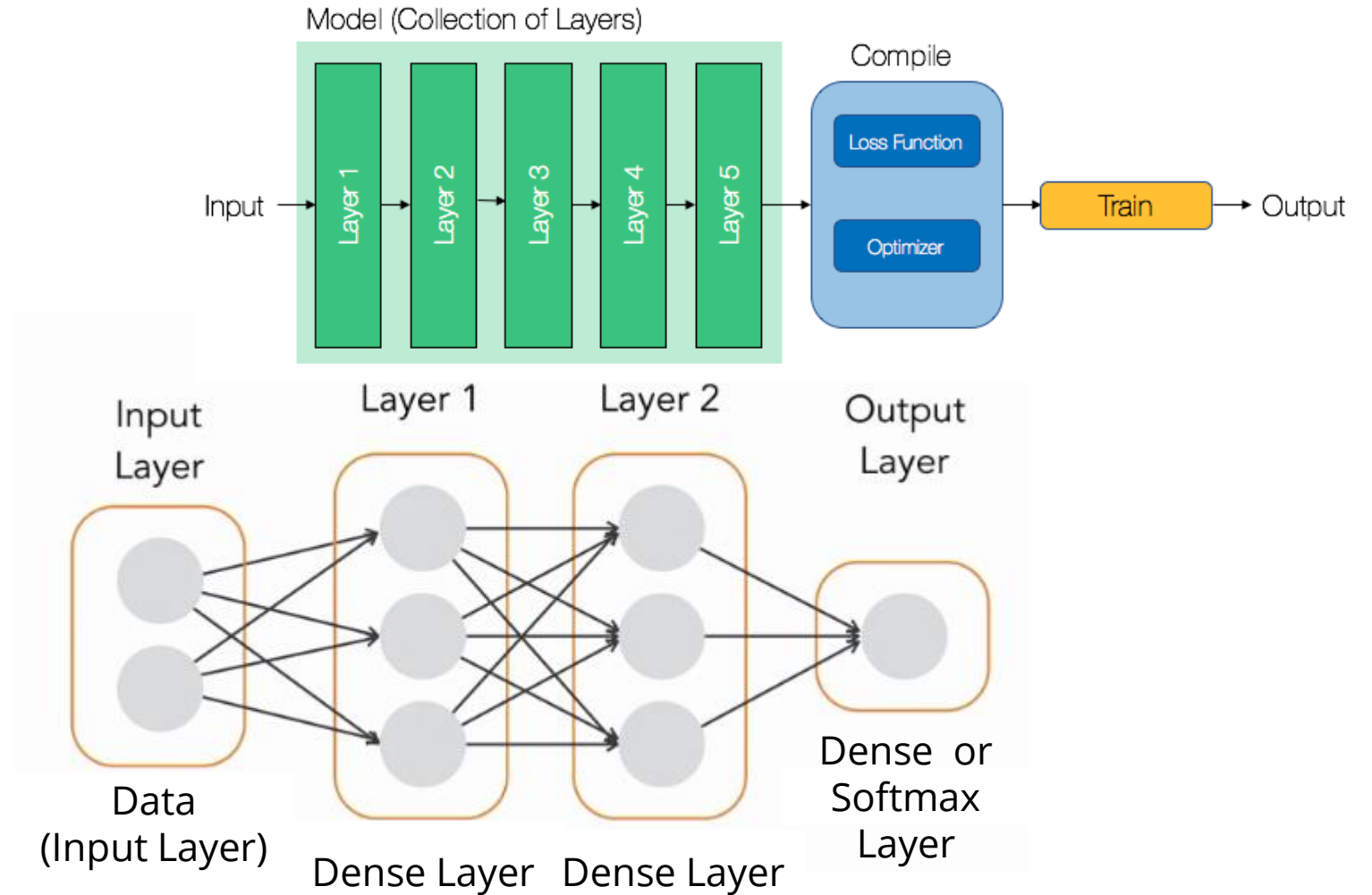
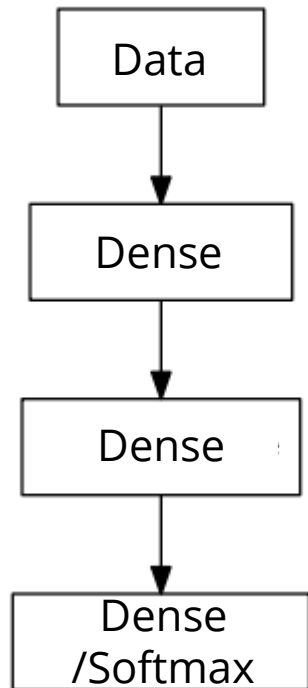
It inherits neural networks functions from packages like tensorflow (Google) or torch (Facebook).



Keras Sequential API

Lets classify cloths (10 classes)!

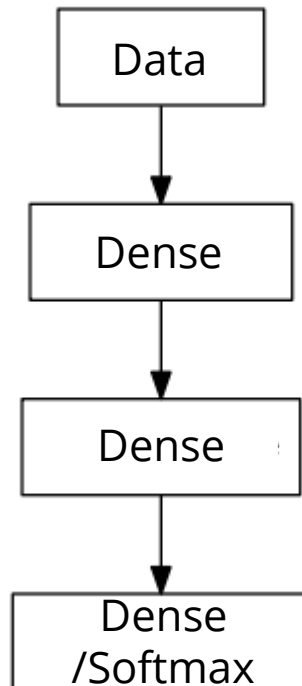
Sequential API



Keras Sequential API

Lets classify cloths (10 classes)!

Sequential API



Load Data:

```
fashion_mnist = keras.datasets.fashion_mnist

(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

Define Model:

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])
```

Define Optimizer:

```
model.compile(optimizer='Adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

Train the Model

```
model.fit(train_images, train_labels, epochs=5)
```

Test the Model

```
test_loss, test_acc = model.evaluate(test_images, test_labels)

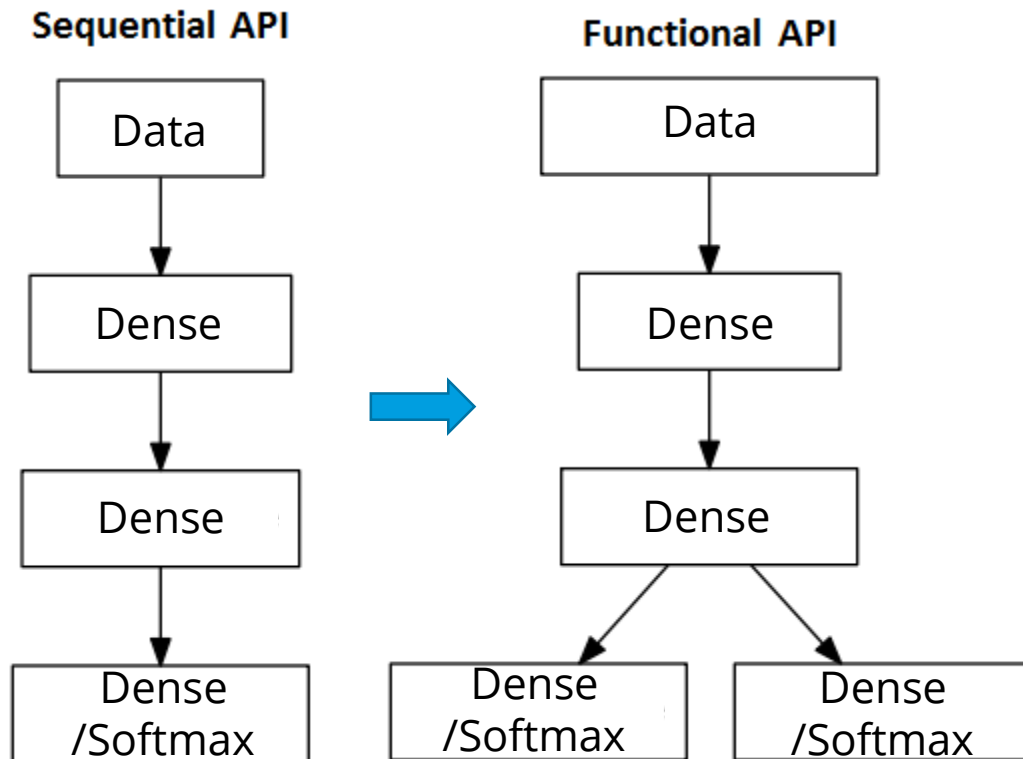
print('Test accuracy:', test_acc)
```

Use model to make predictions:

```
predictions = model.predict(test_images)
```

Keras Functional API

Let's classify cloth and if it's female or male clothing with the same **neural network**! We gonna need **branching** for this which is part of the more flexible **Functional API**.



Define Model:

```
# input layer
data = Input(shape=(28,28))
# feature extraction
hidden1 = Dense(128, activation='relu')(data)
hidden2 = Dense(128, activation='relu')(hidden1)
# classification
class1 = Dense(1, activation='softmax')(hidden2) # male of female clothing?
class2 = Dense(10, activation='softmax')(hidden2) # classification color

# output
model = Model(inputs=data, outputs=[class1, class2])
```

Everything else stays the same!

4. Exercise

Let's train our first classifier with Keras!