



model = Sequential()

model.add(Dense(input_dim=2, units=5,
activation="tanh", name="dense"))

→ model.add(Dense(units=2, activation='softmax'))

Parameters : W, b ?

Minimize a loss function

$$\mathcal{Z} = \{ \underbrace{(x_i, y_i)}_{2D}, \underbrace{C_i}_{\text{class}}, 1 \leq i \leq n \}$$

$$i \quad \underline{L}(C_i, f_{W,b}((x_i, y_i)))$$

Objective: Find (w, b) such that

$\sum_i \underline{L}(c_i, f_{w,b}(x_i, y_i))$ is minimum

Classification \rightarrow Binary Cross-entropy \checkmark

how to minimize?

\rightarrow Optimization procedure

\rightarrow SGD ADAM, ...

model.compile (loss='binary_crossentropy',
optimizer=SGD())

model.fit (train_data, train_labels, epochs=1000,
batch_size=100)

\rightarrow w, b are fixed.

1/ Define the model

model = sequential()

model.add(...) ...
└

2/ Define the loss + optimizer

model.compile(...)

3/ Train model

model.fit(train_data, train_label, 1000, 100)

→

4 model.predict(x_test)

