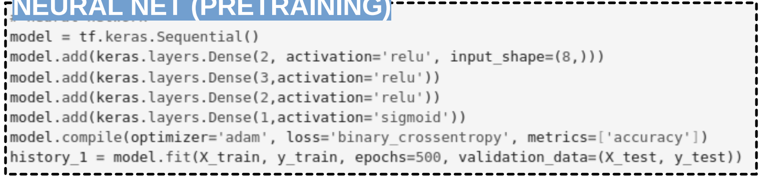
jupyter

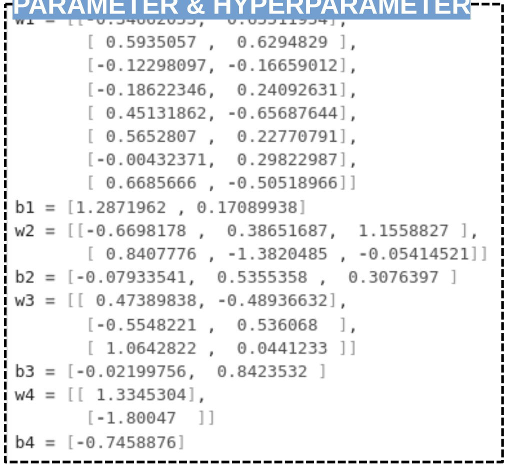
CLOUD & PYTHON

NEURAL NET (PRETRAINING)



```
model = tf.keras.Sequential()
model.add(keras.layers.Dense(2, activation='relu', input_shape=(8,)))
model.add(keras.layers.Dense(3,activation='relu'))
model.add(keras.layers.Dense(2,activation='relu'))
model.add(keras.layers.Dense(1,activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
history_1 = model.fit(X_train, y_train, epochs=500, validation_data=(X_test, y_test))
```

PARAMETER & HYPERPARAMETER

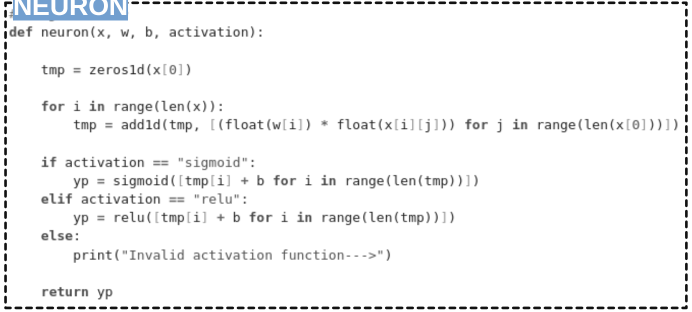


```
w1 = [[-0.34662033,  0.63511934],
       [ 0.5935057 ,  0.6294829 ],
       [-0.12298097, -0.16659012],
       [-0.18622346,  0.24092631],
       [ 0.45131862, -0.65687644],
       [ 0.5652807 ,  0.22770791],
       [-0.00432371,  0.29822987],
       [ 0.6685666 , -0.50518966]]
b1 = [1.2871962 , 0.17089938]
w2 = [[-0.6698178 ,  0.38651687,  1.1558827 ],
       [ 0.8407776 , -1.3820485 , -0.05414521]]
b2 = [-0.07933541,  0.5355358 ,  0.3076397 ]
w3 = [[ 0.47389838, -0.48936632],
       [-0.5548221 ,  0.536068  ],
       [ 1.0642822 ,  0.0441233 ]]
b3 = [-0.02199756,  0.8423532 ]
w4 = [[ 1.3345304],
       [-1.80047  ]]
b4 = [-0.7458876]
```

TRANSFER

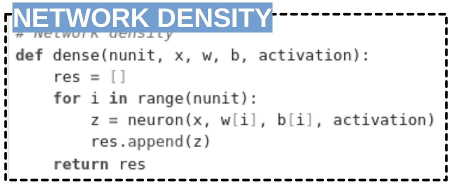
MICROPYTHON

NEURON



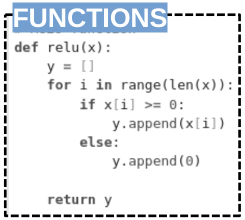
```
def neuron(x, w, b, activation):

    tmp = zeros1d(x[0])

    for i in range(len(x)):
        tmp = add1d(tmp, [(float(w[i]) * float(x[i][j])) for j in range(len(x[0]))])

    if activation == "sigmoid":
        yp = sigmoid([tmp[i] + b for i in range(len(tmp))])
    elif activation == "relu":
        yp = relu([tmp[i] + b for i in range(len(tmp))])
    else:
        print("Invalid activation function--->")

    return yp
```

NETWORK DENSITY



```
# Network density
def dense(nunit, x, w, b, activation):
    res = []
    for i in range(nunit):
        z = neuron(x, w[i], b[i], activation)
        res.append(z)
    return res
```

FUNCTIONS



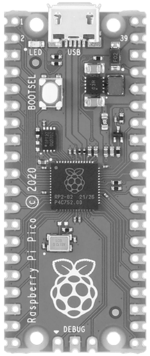
```
def relu(x):
    y = []
    for i in range(len(x)):
        if x[i] >= 0:
            y.append(x[i])
        else:
            y.append(0)

    return y
```

TRANSFER

Raspberry Pi

MICROCONTROLLER

PI PICO



KI-ENNA

www.statistical-thinking.de
Prof. Dr. habil. Dennis Klinkhammer